Datatieteen maisteriohjelma

# Logistisen lämpötiladatan analyysi klusteroinnilla

Selina Lehtoranta

17.4.2020

Helsingin Yliopisto
Matemaattis-luonnontieteellinen tiedekunta

PL 68 (Pietari Kalmin katu 5)
00014 Helsingin yliopisto

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

| Tiedekunta — Fakultet — Faculty | Koulutusohjelma — Utbildningsprogram — Degree programme |
|---|---|
| Matemaattis-luonnontieteellinen tiedekunta | Datatieteen maisteriohjelma |

Tekijä — Författare — Author

Selina Lehtoranta

Työn nimi — Arbetets titel — Title

Logistisen lämpötiladatan analyysi klusteroinnilla

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidantal — Number of pages |
|---|---|---|
| Pro gradu -tutkielma | 17.4.2020 | 42 |

Tiivistelmä — Referat — Abstract

Tutkielma on toteutettu suomalaisen elintarvike- ja logistiikkayrityksen pyynnöstä, ja heidän pääasiallisena tavoitteena on saada vastaus kysymykseen "Voidaanko toimitusketjun lämpötilaa soveltaa toimitusasiakkaan velvoittamaan vastaanotto-lämpötilan mittaukseen?"Tutkielmassa esitetään ja sovelletaan kahta eri klusterointitekniikkaa, jotka ovat $k$-means -klusterointi ja EM-algoritmi Gaussin sekoitemalleille.

Tutkielmassa hieman verrataan näitä kahta klusterointitekniikkaa ja selvitetään, kumpi niistä on parempi tällaisessa tutkimuksessa. Perinteisen EM-GMM lähestymistavan lisäksi EM-algoritmia sovelletaan Gaussin sekoitemalleille hyödyntäen pääkomponenttianalyysia. Näiden lisäksi vastataan tutkimuskysymykseen käyttäen suhteellista muutosta.

Avainsanat — Nyckelord — Keywords

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

# Sisältö

# 1. Johdanto

Tutkielma on toteutettu toimeksiantona yhteistyössä suomalaisen elintarvike- ja logistiikkayrityksen kanssa, ja aiheena on logistiikka ja toimitusketju. Tarkemmin sanoen kyseinen yritys haluaa tutkimuksen toimitusketjun lämpötilojen mittaamisesta ja niiden analysoinnista, aineisto sisältää lämpötilatietojen lisäksi muun muassa aikaleimoja sekä paikkatietoja. Tutkielmassa esitetään ja sovelletaan klusterointitekniikkaa nimeltä $k$-means -klusterointi, sekä odotusarvon maksimointi (Expectation-maximization) algoritmia yhdessä Gaussin sekoitemallien (Gaussian mixture models) kanssa.

Gaussin sekoitemalli (GMM) on parametrinen todennäköisyystiheysfunktio, joka esitetään normaalijakautuneiden komponenttitiheyksien painotettuna summana [4]. GMM:a käytetään usein jatkuvien mittausten tai biometrisen järjestelmän ominaisuuksien todennäköisyysjakauman parametrisena mallina. Parametreja arvioidaan harjoitusaineistosta muun muassa käyttäen iteratiivista odotusarvon maksimointialgoritmia (EM-algoritmi).

Tutkielmassa käytettävä $k$-means -klusterointi on klusterointitekniikka, jota yleisesti käytetään jakamaan aineisto $k$:hon ryhmään automaattisesti [7]. Nimeä $k$-means kyseiselle tekniikalle käytti ensimmäisen kerran James MacQueen vuonna 1967. Klusterointi aloitetaan valitsemalla $k$ klusterikeskusta ja sen jälkeen jalostetaan niitä sillä tavalla, että ensiksi jokainen datapiste $d_i$ osoitetaan sen lähimmälle klusterikeskukselle, jonka jälkeen jokainen klusterikeskus $C_j$ päivitetään sen sisältävien datapisteiden keskiarvoksi. Algoritmi konvergoituu, kun datapisteiden osoittamisessa klustereihin ei ole enää muutoksia.

Odotusarvon maksimointi algoritmi (EM-algoritmi) on käytetyin sekoite uskottavuuden (mixture likelihood) lähestymistapa klusterointiin [8]. EM-algoritmi Gaussian sekoitemalleille on melko herkkä alkuarvoille ja sen komponenttien lukumäärä on annettava etukäteen. Ratkaistakseen nämä EM-algoritmin varjopuolet, on luotu vankka EM-klusterointi algoritmi Gaussin sekoitemalleille, joka antaa uuden tavan ratkaista nämä ongelmat. Kyseisen algoritmin avulla saadaan automaattisesti optimaalinen määrä klustereita.

Tutkimuskysymyksiä, joihin tutkielmassa keskitytään on yhteensä viisi. Voidaanko toimitusketjun lämpötilaa soveltaa toimitusasiakkaan velvoittamaan vastaanottolämpötilan mittaamiseen? Millä tavalla toimitusketjun lämpötilat korreloivat mittausajan

ja erilaisten poikkeamien kanssa? Onko jokin mittauspisteistä tai mittausvälineistä turha? Miten lämpötilat x ja y muuttuvat lämpötilan z muuttuessa? Onko kaluston iällä vaikutusta lämpötiloihin?

Toisessa kappaleessa keskitytään tutkielmassa käytettyihin metodologioihin, joihin kuuluu toimitusketju, ohjaamaton oppiminen, klusterointi, Gaussin sekoitemalli, pääkomponenttianalyysi sekä suhteellinen muutos. Kolmannessa kappaleessa esitetään tutkimuksen tulokset ja viimeisessä kappaleessa on tuloksien pohdintaa.

# 2. Metodologia

## 2.1 Toimitusketju

Toimitusketju on useista työvaiheista, yrityksistä sekä henkilöistä koostuva tapahtumien sarja, jossa edetään aina tuotteen tai palvelun raaka-aineiden tuotannosta niiden lopulliseen muotoon ja lopulta toimitukseen kuluttajalle [5]. Toimitusketjun tehtävänä on muodostaa raaka-aineista kuluttajan tilaama tuote tai palvelu ja huolehtia, että kuluttaja vastaanottaa tilaamansa tuotteen tai palvelun. Näin ollen toimitusketju pitää sisällään materiaali-, tieto- ja rahavirrat. Tässä tutkielmassa keskitymme kuitenkin toimitusketjuun, joka koostuu myyntitilausprosessista, logistisesta prosessista sekä laskutusprosessista. Myyntitilausprosessissa kuluttaja tilaa haluamansa tuotteen tai palvelun paikan päältä, puhelimitse, verkkokaupan kautta tai sähköisesti käyttäen EDI-tietoliikennesanomaa tai API-rajapintaa. Logistinen prosessi koostuu varastoprosessista (sisälogistiikka), irtoprosessista (siirtologistiikka) sekä jakeluprosessista (jakelulogistiikka). Viimeinen tutkielmassa käsiteltävä toimitusketjun prosessi on Laskutusprosessi, jossa laskutus tapahtuu paperilaskuna, sähköpostilaskuna tai sähköisesti käyttäen EDI-tietoliikennesanomaa tai API-rajapintaa.

Jakeluketjuhallinta (Supply chain management) on toimenpide, joka lisää toimitusketjun asiakaslähtöisyyttä ja kustannustehokkuutta [5]. Asiakaslähtöisyyden sekä kustannustehokkuuden tulisi kulkea käsi kädessä, tällöin päädytään haluttuun lopputulokseen. Kuluttajan tarpeet ja toiveet on otettava huomioon, mutta samalla on muistettava kustanustehokkuus. Unohtamalla asiakaslähtöisyys, saavutetaan ei-toivottu tila kuluttajan näkökulmasta.

Yritysten valmistamat tuotteet toimitetaan asiakkaille ja lopulta kuluttajille, tätä kutsutaan myös työntämiseksi [5]. Tuotteet on valmistettu kuulematta asiakkaiden toiveita ja tällöin tämä työntöohjaus aikaansaa yli- tai alivarastoja. Työntöohjauksen vastakohta on imuohjaus, jossa tuote valmistetaan asiakkaan toiveiden määrittämisen ja tilauksen jälkeen, kun toimintaprosessi käynnistyy. Tätä ajatusmallia kutsutaan kysyntäketjuhallinnaksi (Demand chain management), joka korostaa asiakaslähtöisyyttä. Sopivin toimintamalli olisi näiden työntö- ja imuohjauksien käyttö rinnakkain samassa toimitusketjussa. Toimitusketjun alkupään vakiokomponenttien ja moduulien kustannustehokas työntöpe-

rusteinen massatuotanto ja hankinta lisäävät toimitusketjun sisäistä tehokkuutta. Toimitusketjun loppupään imuohjauksessa kuluttajan toiveiden mukaan suoritettu kokoonpano taas lisää toimitusketjun asiakaslähtöisyyttä.

## 2.2 Ohjaamaton oppiminen

Ohjaamattomassa oppimisessa on syötteitä, mutta ei ohjattuja tulosteita, tästä huolimatta aineistosta voidaan oppia suhteista ja rakenteesta [2]. Yleisimmät tilastolliset oppimisen ongelmat jakautuvat kahteen ryhmään: ohjattuun ja ohjaamattomaan oppimiseen. Ohjaamattoman oppimisen tilanteessa jokaiselle havainnolle $i = 1, \ldots, n$ havaitaan mittausvektori $x_i$, mutta ei siihen liittyvää vastemuuttujaa $y_i$. Lineaarista regressiomallia ei voida sovittaa tässä tilanteessa, sillä vastemuuttuja ei ole ennustettavana. Ohjaamattoman tilanteesta tekee se, että analyysia valvova vastemuuttuja puuttuu.

Ohjaamaton oppiminen on joukko tilastollisia työkaluja, jotka on tarkoitettu ympäristöön, jossa on vain joukko ominaisuuksia $X_1, X_2, \ldots, X_p$ mitattuna $n$ havainnolla [2]. Ennustamisesta ei olla kiinnostuneita, sillä tiedossa ei ole liittyvää vastemuuttujaa $Y$. Pikemminkin ollaan kiinnostuneita löytämään mielenkiintoisia asioita mittauksista ominaisuuksilla $X_1, X_2, \ldots, X_p$. Onko esimerkiksi informatiivista tapaa visualisoida aineistoa? Tai voidaanko muuttujien ja havaintojen joukosta löytää alaryhmiä? Ohjaamaton oppiminen viittaa monipuoliseen joukkoon tilastollisia tekniikoita, joiden avulla vastataan muunmuassa edellä esitetyihin kysymyksiin. Yksi ohjaamattoman oppimisen tekniikka on klusterointi, joka on laaja luokka menetelmiä, joilla etsitään tuntemattomia osajoukkoja aineistosta.

Ohjaamaton oppiminen on usein paljon monimutkaisempi kuin ohjattu oppiminen [2]. Sen käyttö on yleensä paljon subjektiivisempaa, eikä analyysille ole yksinkertaista tavoitetta, kuten vasteen ennustaminen. Ohjaamaton oppiminen suoritetaan yleensä osana tutkivaa data-analyysiä. Lisäksi ohjaamattoman oppimisen menetelmistä saatujen tuloksien arvioiminen on vaikeaa, sillä ei ole olemassa yleisesti hyväksyttyä mekanismia ristiinvalidoinnin tai riippumattoman aineiston tuloksien validoinnin suorittamiseen. Tähän eroon löytyy yksinkertainen syy, jos ennustavaa mallia sovitetaan käyttäen ohjatun oppimisen tekniikkaa, on mahdollista tarkistaa tehty työ katsomalla miten hyvin kyseinen malli ennustaa vastauksen $Y$ havainnoilla, joita ei olla sovitettu malliin. Ohjaamattomassa oppimisessa ei kuitenkaan ole tapaa tarkistaa tehtyä työtä, sillä oikeaa vastausta ei tiedetä ja näin ollen ongelma on ohjaamaton.

Ohjaamattomien oppimistekniikoiden merkitys kasvaa yhä monilla aloilla [2]. Esimerkiksi syöpätutkimuksessa saatetaan määrittää geeniekspressitasot rintasyöpäpotilailla, ja etsiä alaryhmiä rintasyöpänäytteistä tai -geeneistä, täten saadaan parempi käsitys taudista. Verkkokauppasivustolla voidaan yrittää tunnistaa samankaltaisen selaus- ja os-

toshistorian omaavien ostajien ryhmiä, samoin kuin kunkin ryhmän ostajille erityisen kiinnostavia tavaroita. Tällöin yksittäiselle ostajalle pystytään suositellusti näyttämään tavaroita, joista hän on erityisen todennäköisesti kiinnostunut, samankaltaisten ostajien selaus- ja ostohistorian perusteella. Hakukone voi valita mitä hakutuloksia näytetään tietyille yksilöille muiden yksilöiden, joilla on samanlaiset hakutekniikat, klikkausten perusteella. Nämä tilastollisen oppimisen tehtävät sekä monet muut voidaan suorittaa käyttäen ohjaamattoman oppimisen tekniikoita.

## 2.3 Klusterointi

Klusterointi viittaa laajaan joukkoon tekniikoita, joilla etsitään aineistosta alaryhmiä eli klustereita [2]. Kun aineistosta saatuja havaintoja klusteroidaan, pyritään ne jakamaan erillisiin samankaltaisia havaintoja sisältäviin ryhmiin. Havainnot eri ryhmissä voivat olla hyvinkin erilaisia toistensa kanssa. Klusteroiminen vaatii kuitenkin, että määritetään mitä kahden tai useamman havainnon samankaltaisuus tai erilaisuus tarkoittaa. Tämä on usein aihekohtaista ja määrittely on tehtävä tutkittavan aineiston perusteella.

Klusterointi on ohjaamattoman oppimisen ongelma, sillä siinä yritetään etsiä rakennetta aineiston perusteella [2]. Ohjatun oppimisen ongelman tavoitteena on yrittää ennustaa tulosvektori, esimerkiksi selviytymisaika tai vaste lääkehoitoon. Klusteroinnilla pyritään yksinkertaistamaan aineistoa vähällä määrällä yhteenvetoja, ja se etsii homogeenisiä osaryhmiä havaintojen joukosta.

Yleisesti havaintoja pystytään klusteroimaan ominaisuuksien perusteella, että niiden joukosta pystyy löytämään osaryhmät [2]. Klusterointi on suosittua monella alalla ja sen vuoksi on olemassa useita klusterointitekniikoita, joista esimerkkinä $k$-means -klusterointi, joka on yksi tunnetuimmista klusterointitekniikoista. Siinä jaetaan havainnot ennaltamäärättyyn määrään klustereita.

Klusterointi voi olla hyvin hyödyllinen työkalu aineiston analysointiin ohjaamattomassa oppimisessa [2]. Klusteroinnissa löytyy kuitenkin myös paljon ongelmia, muun muassa pienet päätökset, joilla on suuria seurauksia; saatujen klustereiden validointi; muut näkökohdat klusteroinnissa sekä karkaistu lähestymistapa klusteroinnin tulosten tulkintaan.

Jotta klusterointi pystytään toteuttamaan, on tehtävä muutamia valintoja [2]. On muun muassa pohdittava, että pitääkö havainnot tai ominaisuudet aluksi standardoida jollain tavalla. Pitäisikö esimerkiksi muuttujilla olla keskiarvona 0 ja pitäiskö ne skaalauta siten, että saadaan keskihajonnaksi 1. Ongelmana $k$-means klusteroinnissa on myös se, että kuinka monta klusteria aineistosta pitäisi valita. Näillä molemmilla valinnoilla on suuri vaikutus saavutettuun tulokseen. Käytännössä useita valintoja testataan ja niistä valitaan se, mikä on käytännöllisin tai tulkitsevin. Näitä metodeita käyttäen ei ole yhtä oikeaa

ratkaisua, vaan kaikkia tuloksia, jotka paljastaa kiinnostavia näkökohtia aineistosta, tulisi harkita.

Joka kerta, kun klusterointi suoritetaan aineistolle, löydetään klustereita [2]. Halutaan kuitenkin olla varmoja siitä, että onko saadut klusterit oikeasti osajoukkoja aineistosta, vai ovatko ne vain tuloksia "kohinan" klusteroinnista. Esimerkiksi jos saataisiin riippumaton joukko havaintoja, näyttäisikö nämä samat havainnot saman joukon klustereita? On olemassa monia tekniikoita, joilla määritetään p-arvo klusterille, tämän avulla arvioidaan onko klusterissa enemmän todisteita, kuin mitä voisi olettaa. Ei kuitenkaan ole yhteisymmärrystä yhdestä parhaasta lähestymistavasta.

Jokainen havainto määritetään klusteriin $k$-means klusteroinnissa [2]. Kuitenkin joinain kertoina tämä ei ole soveliasta. Esimerkiksi, oletetaan, että useimmat havainnot kuuluvat pieneen määrään tuntemattomia osaryhmiä, ja pieni osajoukko havainnoista eroaa toisistaan ja kaikista muista havainnoista. Kun $k$-means -klusterointi pakottaa jokaisen havainnon klusteriin, löydetyt klusterit saattavat olla voimakkaasti vääristyneitä mihinkään klusteriin kuulumattomien poikkeamien vuoksi. Tällaisten poikkeamien läsnäolo pystytään hyvin ottamaan huomioon sekoitemalleilla (Mixture models). Lisäksi klusterointitekniikat ovat usein herkkiä aineiston häiriöille. Esimerkiksi, oletetaan, että klusteroidaan $n$ havaintoa ja sitten klusteroidaan havainnot uudelleen satunnaisen $n$:n osajoukon havaintojen poistamisen jälkeen. Voisi olettaa ja toivoa, että näin saadut kaksi ryhmää ovat samanlaisia, mutta näin ei kuitenkaan usein ole.

Klusterointiin liittyy ongelmia, mutta se voi myös olla todella hyödyllinen ja pätevä tilastollinen työkalu jos sitä käytetään oikein [2]. Pienillä päätöksillä, esimerkiksi, että miten klusterointi suoritetaan, ja miten aineisto standardoidaan ja minkä tyylistä linkitystä käytetään, voi olla suuri merkitys tulokseen. Tämän vuoksi klusterointi suositellaan suorittamaan näiden parametrien erilaisilla valinnoilla, ja katsomaan kokonaista joukkoa tuloksia. Tällöin nähdään mitä kuvioita ja malleja jatkuvasti esiintyy. Koska klusterointi voi olla epävakaa, suositellaan aineiston osajoukkojen klusterointia, jotta saadaan käsitys klusterien kestävyydestä. On tärkeää olla tarkka siitä, kuinka klusterointianalyysin tulokset raportoidaan. Näitä tuloksia ei pitäisi pitää aineiston absoluuttisina totuuksina, vaan niitä pitäisi pitää tieteellisen hypoteesin ja jatkotutkimuksen kehittämisen lähtökohtana, mieluiten riippumattomassa aineistossa.

### 2.3.1 $K$-means -klusterointi

Klusterointimenetelmä $k$-means -klusterointi jakaa automaattisesti aineiston ennaltamäärättyyn lukumäärään ($k$) erillisiä klustereita [2, 7]. Käyttääkseen $k$-means -klusterointia, aluksi täytyy määritellä $k$ eli haluttu klustereiden lukumäärä. Tämän jälkeen $k$-means algoritmi osoittaa jokaisen havainnon yhdelle $k$:sta klusterista.

Tämä klusterointitekniikka perustuu yksinkertaiseen ja intuitiiviseen matemaattiseen ongelmaan [2]. Olkoot $\{1, 2, \ldots, n\}$ klusteroitava joukko ja $C_1, \ldots C_k$ sarjoja, jotka sisältävät havaintojen indeksit kussakin klusterissa. Nämä sarjat täyttävät seuraavat ominaisuudet:

- $C_1 \cup C_2 \cup \ldots \cup C_k = \{1, \ldots, n\}$, eli jokainen havainto kuuluu ainakin yhteen $k$:sta klusterista
- $C_k \cap C_{k'} = \emptyset$ kaikille $k \neq k'$, eli havainnot eivät kuulu kuin yhteen klusteriin ja näin ollen klusterit ovat pistevieraita.

Esimerkiksi jos havainto $i$ kuuluu klusteriin $k$, niin merkitään $i \in C_k$. Hyvä klusterointi on sellainen, että klusterin sisäinen variaatio on mahdollisimman pieni. Klusterin $C_k$ sisäinen variaatio on mitta $W(C_k)$ määrästä, jolla klustereiden sisäiset havainnot eroavat. Näin ollen haluamme ratkaista seuraavan yhtälön:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{k} W(C_k) \right\}. \tag{2.1}$$

Tämä yhtälö tarkoittaa sitä, että haluamme jakaa havainnot klustereihin siten, että klustereiden sisäinen variaatio summattuna yli kaikkien klustereiden pysyy mahdollisimman pienenä.

Yhtälön (2.1) ratkaiseminen vaikuttaa kohtuulliselta, mutta, jotta siitä saadaan toimintakelpoinen, on klustereiden sisäinen variaatio määriteltävä [2]. Yleisin tapa määrittelemiseen on käyttää euklidisen etäisyyden neliötä:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{P} (x_{ij} - x_{i'j})^2, \tag{2.2}$$

missä $|C_k|$ on havaintojen määrä $k$:nnessa klusterissa. $K$:nnen klusterin sisäinen variaatio on klusterin $k$ kaikkien havaintojen pareittain neliöityjen euklidisten etäisyyksien summa jaettuna kaikkien klusterin $k$ havaintojen kokonaismäärällä. Yhdistämällä nämä yhtälöt, saadaan optimointiongelma, joka määrittää $k$-means klusteroinnin:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{P} (x_{ij} - x_{i'j})^2 \right\}. \tag{2.3}$$

Yhtälön ratkaisemiseksi tarvitaan algoritmi eli metodi, joka jakaa havainnot $k$:hon klusteriin siten, että yhtälön tavoite (objective) minimoidaan [2]. Tämä on vaikea ongelma, sillä on olemassa melkein $k^n$ tapaa jakaa $n$ havaintoa $k$:hon klusteriin. Määrästä tulee iso, elleivät $k$ ja $n$ ole pieniä. On kuitenkin olemassa yksinkertainen algoritmi, jonka voidaan osoittaa tarjoavan paikallisen optimin yhtälölle (2.3). Kyseinen algoritmi on seuraavanlainen:

- Ensiksi osoitetaan satunnaisesti luku 1:stä $k$:hon jokaiselle havainnolle. Nämä luvut ovat havaintojen alkukeskeiset klusteritehtävät (cluster assignment).
- Seuraavaksi iteroidaan, kunnes klusteritehtävät eivät enää muutu. Tällöin jokaiselle klusterille $k$ lasketaan klusterin painopiste, $k$:nnen klusterin painopiste on $p$:n ominaisarvon vektori $k$:nnen klusterin havainnoille.
- Tämän jälkeen jokainen havainto osoitetaan klusteriin, jonka painopiste on lähimpänä. Etäisyys mitataan yhtälöllä (2.2).

Algoritmi pienentää tavoitteiden arvoa jokaisella askeleella [2]. Tämän voi ymmärtää seuraavan yhtälön avulla:

$$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2, \tag{2.4}$$

jossa $\bar{x}_{kj} = \frac{1}{|C_k|}\sum_{i\in C_k} x_{ij}$ on ominaisuuden $j$ keskiarvo klusterissa $C_k$. Algoritmissa jokaisen ominaisuuden klusterikeskiarvot ovat vakioita, jotka minimoivat neliösumman poikkeamia. Tämän lisäksi havaintojen uudelleenjako voi vain parantaa esitettyä yhtälöä. Näin ollen algoritmin suorittamisen jälkeen saatu klusterointi paranee siihen asti, kunnes tulos ei enää muutu, ja optimointiongelma ei koskaan kasva. Kun tulos ei enää muutu, tällöin ollaan saavutettu lokaali optimi eli paras mahdollinen tulos. Algoritmissa klusterin painopiste lasketaan jokaiselle klusterille osoitettujen havaintojen keskiarvoista ja tästä tuleekin $k$-means klusteroinnin nimi.

Koska $k$-means algoritmi löytää lokaalin, eikä globaalia optimia, saadut tulokset riippuvat jokaisen havainnon alkuperäisestä satunnaisesta klusteritehtävästä [2]. Tämän vuoksi on tärkeää suorittaa algoritmi useaan otteeseen erilaisilla satunnaisilla konfiguraatioilla. Tämän jälkeen valitaan paras ratkaisu, jolla esimerkiksi optimointiongelma olisi pienin.

Suorittaakseen $k$-means klusteroinnin, on valittava kuinka monta klusteria odotamme aineistosta, tämä on yksi $k$-means klusteroinnin heikkous [2]. Klusterien määrän, eli arvon $k$ valitseminen ei ole kuitenkaan helppoa.

## 2.4 Gaussin sekoitemalli

Gaussin sekoitemalli (GMM) on parametrinen todennäköisyystiheysfunktio, joka esitetään Gaussin komponenttitiheyksien painotettuna summana [4]. GMM:a käytetään usein jatkuvien mittausten tai biometrisen järjestelmän ominaisuuksien todennäköisyysjakauman parametrisena mallina. Parametreja arvioidaan harjoitus aineistosta muun muassa käyttäen iteratiivista odotusarvon maksimointi algoritmia (EM-algoritmi).

Gaussin sekoitemalli on seuraavan yhtälön antama painotettu summa $M$-komponentin Gaussin tiheyksistä:

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} w_i g(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{2.5}$$

jossa $\boldsymbol{x}$ on $D$ ulotteinen jatkuva-arvoinen aineistovektori, $w_i = 1, \ldots, M$ ovat sekoitepainoja, ja $g(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), i = 1, \ldots, M$ ovat komponentin Gaussin tiheydet [4]. Jokaisen komponentin tiheys on monimuuttuja Gaussin yhtälössä, joka on muotoa

$$g(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)'\Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i)\right\}, \tag{2.6}$$

jossa $\boldsymbol{\mu}_i$ on keskiarvovektori ja $\Sigma_i$ on kovarianssimatriisi. Seospainot täyttävät seuraavan vaatimuksen $\sum_{i=1}^{M} w_i = 1$.

Täydellisessä Gaussin sekoitemallissa on parametreina keskiarvovektorit, kovarianssimatriisit sekä sekoitepainot kaikkien komponenttien tiheyksistä [4]. Nämä parametrit esitetään yhdessä merkinnällä

$$\lambda = \{w_i, \boldsymbol{\mu}_i, \Sigma_i\} \qquad i = 1, \ldots, M. \tag{2.7}$$

Yhtälössä (2.7) esitetyssä GMM:ssä on useita vaihtoehtoja. Muun muassa kovarianssimatriisi $\Sigma_i$ voi olla täydellinen, tai rajoitettu diagonaaliseksi. Lisäksi parametrejä voidaan jakaa tai sitoa Gaussin komponentteihin, esimerkiksi kaikilla komponenteilla voi olla sama kovarianssimatriisi. Mallimääritysten valinta on usein määritetty GMM parametrien arvioimiseksi käytettävissä olevan aineiston perusteella ja kuinka GMM:ää käytetään tietyssä biometrisessä sovelluksessa.

On myös tärkeää huomata, että koska Gaussin komponentit toimivat yhdessä mallintaakseen ominaisuuksien kokonaistiheyden, täydellinen kovarianssimatriisi ei ole välttämätön vaikka ominaisuudet eivät olisi tilastollisesti riippumattomia [4]. Diagonaalisen kovarianssimatriisin lineaarikombinaatio pystyy mallintamaan ominaisvektorielementtien väliset korrelaatiot. Käyttämällä suurempaa joukkoa diagonaalisia kovarianssimatriiseja, voidaan saada sama vaikutus kuin käyttämällä $M$ suuruista joukkoa täysiä kovarianssimatriiseja.

Gaussin sekoitemalleja käytetään usein biometrisissä järjestelmissä, etenkin puhujan tunnistamisjärjestelmissä, sillä ne pystyvät edustamaan suurta otosjakaumien luokkaa [4]. Yksi GMM:n tehokkaista ominaisuuksista on sen kyky muodostaa tasaisia arvioita mielivaltaisesti muotoilluille tiheyksille. Klassinen unimodaalinen (uni modal) Gaussin malli edustaa ominaisuusjakaumia paikan, elliptisen muodon sekä vektorikvantisoijan (vector quantizer) avulla, tai lähin naapurimalli edustaa jakaumaa diskreetin ominaismallijoukon avulla. GMM toimii hybridinä näiden kahden mallin välillä käyttäen diskreettiä joukkoa

Gaussin funktioita, joilla kaikilla on oma keskiarvo ja kovarianssimatriisi, mahdollistaakseen paremman mallinnuksen valmiuden.

GMM:n käyttö ominaisuusjakaumien esittämiseen biometrisessä järjestelmässä voi olla motivoitunut myös intuitivisella käsityksellä siitä, että yksittäisten komponenttitiheyksien avulla voidaan mallintaa jotakin taustalla olevaa piilotettujen luokkien joukkoa [4]. Esimerkiksi puhujan tunnistamisessa, on järkevää olettaa, että akustisen tilan spektriin liittyvät piirteet vastaavat puhujan laajoista foneettisista tapahtumista, kuten vokaalisoinnuista, nasaaleista sekä hankausäänteistä. Nämä akustiset luokat heijastavat joitain yleisiä puhujasta riippuvia ääniratojen kokoonpanoja, jotka ovat hyödyllisiä puhujaidentiteetin karakterisoinnissa. Akustisen luokan $i$ spektrimuoto voidaan esittää $i$:nnen komponenttitiheyden keskiarvolla $\boldsymbol{\mu}_i$, ja keskimääräisen spektrimuodon vaihtelu voidaan esittää kovarianssimatriisilla $\Sigma_i$. Koska kaikki GMM:n kouluttamiseen käytetyt ominaisuudet ovat merkitsemättömiä ja havaintojen luokkaa ei tunneta, akustiset luokat ovat piilossa. GMM:ää voidaan myös katsella yksitilaisena Markovin piilomallina, jolla on Gaussin sekoitteen havaintotiheys, tai ergodisella Gaussin havainto Markovin piilomallina, jolla on kiinteät yhtä suuret siirtymätodennäköisyydet. Oletetaan, että ominaisvektorit ovat riippumattomia, jolloin näistä piilotetuista akustisista luokista piirrettyjen ominaisvektorien havaintotiheydet ovat Gaussin sekoite.

### 2.4.1 Odotusarvo ja maksimointi -algoritmi

Sekoitemalleille käytetään usein odotusarvon maksimointialgoritmi (EM-algoritmi) [8]. EM-algoritmi on melko herkkä alkuarvoille, joissa komponenttien lukumäärä on annettava etukäteen. EM-algoritmien komponenttien lukumäärään liittyvien alkuperäisten arvojen kestävyyteen on kiinnitetty vähemmän huomiota, vaikka on pohdittu EM-algoritmin alkuperäisiä ongelmia sekä harkittu komponenttien lukumäärän arviointia.

Useat eri tutkijat keksivät ja käyttivät EM-algoritmia itsenäisesti, kunnes Arthur Dempster toi heidän ideansa yhteen ja loi termin EM-algoritmi [3]. Tämän jälkeen on julkaistu monia artikkeleita eri aloilta liittyen EM-algoritmiin. EM-algoritmia voidaan käyttää, kun on olemassa taustajoukko, jolla on tunnettu jakaumafunktio, jota tarkkaillaan monesta yhteen -kartoituksen keskiarvojen avulla.

Olkoon aineisto $\{X_1, X_2, \ldots, X_n\}$ $n$ kokoinen satunnaisotos monimuuttuja sekoitemallista, jossa vektorit $x_j$ ovat $d$-ulotteisia.

$$f(x; \alpha, \theta) = \sum_{k=1}^{c} \alpha_k f(x; \theta_k), \tag{2.8}$$

jossa $\alpha_k > 0$ ovat sekoitussuhteet rajoituksella $\sum_{k=1}^{c} \alpha_k = 1$ ja $f(x; \theta_k)$ on $x$:n tiheys luokasta $k$ parametreilla $\theta_k$ [8]. Olkoot $Z = \{Z_1, Z_2, \ldots\}$ puuttuva data, jossa $Z_i \in \{1, 2, \ldots, c\}$. Jos $Z_i = k$, niin $i$:nnes datapiste kuuluu luokkaan $k$. Täten saadaan

täydellisen aineiston $\{X_1, X_2, \ldots, X_n, Z_1, Z_2, \ldots, Z_n\}$ yhteistiheysfunktioksi

$$f(x_1, \ldots, x_n, z_1, \ldots, z_n; \alpha, \theta) = \prod_{i=1}^{n} \prod_{k=1}^{c} [\alpha_k f(x_i; \theta_k)]^{Z_{ki}}, \tag{2.9}$$

jossa

$$z_{ki} = \begin{cases} 1, & \text{jos } Z_i = k \\ 0, & \text{jos } Z_i \neq k. \end{cases}$$

Log uskottavuusfunktio saadaan seuraavasti:

$$L(\alpha, \theta; x_1, \ldots, x_n, z_1, \ldots, z_n) = \sum_{i=1}^{n} \sum_{k=1}^{c} z_{ki} \ln[\alpha_k f(x_i; \theta_k)] \tag{2.10}$$

**Algoritmin E-askel.** Koska piileviä muuttujia $Z_{ki}$ ei tunneta, ehdollinen odotusarvo $E(Z_{ki}|x_i; \alpha, \theta)$ korvaa muuttujat $Z_{ki}$ [8]. Bayesin lauseen mukaan saadaan

$$\hat{z}_{ki} = E(Z_{ki}|x_i; \alpha, \theta) = \frac{\alpha_k f(x_i; \theta_k)}{\sum_{s=1}^{C} \alpha_s f(x_i; \theta_s)} \tag{2.11}$$

**Algoritmin M-askel.** Rajoituksen $\sum_{k=1}^{C} \alpha_k = 1$ mukaan maksimoidaan

$$\tilde{L}(\alpha, \theta; x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{k=1}^{c} \hat{z}_{ki} \ln[\alpha_k f(x_i; \theta_k)] \tag{2.12}$$

Voidaan saada päivitetty yhtälö mittasuhteiden sekoittamiseksi käyttäen parametria

$$\alpha_k = \frac{\sum_{i=1}^{n} \hat{z}_{ki}}{n}. \tag{2.13}$$

Tarkastellaan nyt monimuuttuja Gaussin sekoitemallia, jossa vektorit $x_j$ ovat $d$-ulotteisia

$$\begin{aligned} f(x; \alpha, \theta) &= \sum_{k=1}^{c} \alpha_k f(X; \theta_k) \\ &= \sum_{k=1}^{c} \alpha_k (2\pi)^{(d/2)} |\Sigma_k|^{-(1/2)} e^{-(1/2)(x-\mu_k)' \Sigma_k^{-1} (x-\mu_k)}. \end{aligned} \tag{2.14}$$

Parametri $\theta_k$ koostuu keskiarvovektorista $\mu_k$ ja kovarianssimatriisista $\Sigma_k$ [8]. Täten näiden parametrien päivitetyt yhtälöt ovat seuraavanlaiset:

$$\mu_k = \frac{\sum_{i=1}^{n} \hat{z}_{ki} x_i}{\sum_{i=1}^{n} \hat{z}_{ki}} \tag{2.15}$$

$$\Sigma_k = \frac{\sum_{i=1}^{n} \hat{z}_{ki} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{n} \hat{z}_{ki}} \tag{2.16}$$

Täten EM-klusterointialgoritmi voidaan tiivistää seuraavasti:

**Ensimmäinen askel:** Valitaan $2 \leq c \leq n$ ja $\varepsilon > 0$. Annetaan alkuarvot $\hat{z}^{(0)} =$

$(\hat{z}_1^{(0)}, \ldots, \hat{z}_c^{(0)})$ ja olkoon $s = 1$.

**Toinen askel:** Lasketaan $\alpha^{(s)}$ ja $\mu^s$ arvolla $\hat{z}^{(s-1)}$ käyttäen kaavoja (2.13) ja (2.15).

**Kolmas askel:** Lasketaan $\Sigma^{(s)}$ arvoilla $\hat{z}^{(s-1)}$ ja $\mu^{(s)}$ käyttäen yhtälöä (2.16).

**Neljäs askel:** Päivitetään $\hat{z}^{(s)}$ arvoilla $(\alpha^{(s)}, \mu^{(s)}, \Sigma^{(s)})$ käyttäen yhtälöä (2.11).

**Viides askel:** Verrataan $\hat{z}^{(s)}$ ja $\hat{z}^{(s-1)}$ keskenään sopivassa matriisinormissa $\|\cdot\|$. Jos $\|\hat{z}^{(s)} - \hat{z}^{(s-1)}\| < \varepsilon$, lopetetaan, muulloin $s = s + 1$ ja palataan toiseen askeleeseen.

EM-algoritmi on melko herkkä alustukselle, joten klusterien lukumäärä pitäisi antaa etukäteen [8]. Klusterien lukumäärää ja sekoitemallin parametrien estimoimiseen samanaikaisesti on luotu algoritmi, joka käyttää viestin minimipituuskriteerin (minimum message length criterion) erityistä muotoa. Tämä kriteeri on seuraavan kustannusfunktion minimointi EM-estimaatin avulla.

$$\begin{aligned} K(\alpha, \theta; x_1, \ldots, x_n) =& \frac{P}{2} \sum_{m:\alpha_m > 0} \ln\left(\frac{n\alpha_m}{12}\right) + \frac{c_{nz}}{2} \ln\left(\frac{n}{12}\right) + \frac{c_{nz}(P+1)}{2} \\ & - \sum_{i=1}^{n} \ln\left[\sum_{k=1}^{c} \alpha_k f(x_i; \theta_k)\right], \end{aligned} \tag{2.17}$$

missä $P$ on kunkin komponentin määrittelevien parametrien lukumäärä ja $c_{nz}$ on komponenttien, joiden todennäköisyys ei ole nolla, lukumäärä. Tällöin päivitetty yhtälö suhteelle on seuraavanlainen:

$$\frac{\alpha_k = \max\{0, \sum_{i=1}^{n} \hat{z}_{ki} - \frac{P}{2}\}}{\sum_{s=1}^{c} \max\{0, \sum_{i=1}^{n} \hat{z}_{si} - \frac{P}{2}\}} \tag{2.18}$$

Gaussin sekoitemallissa monimuuttuja $\theta_k = (\mu_k, \Sigma_k)$, $P = d + (d(d+1)/2)$ ja päivitetyt yhtälöt $\hat{z}_{ki}, \mu_k$ ja $\Sigma_k$ ovat samat kuin yhtälöt (2.11), (2.15) ja (2.16). Parametrien päivittämistä varten on käytetty komponenttikohtaista (component-wise) EM-menetelmää, jossa parametrit päivitetään peräkkäin. Figueiredon ja Jainin esittämä algoritmi suoritetaan ensin syöttämällä suurempia klusterien lukumääriä ja sitten käytetään yhtälöä (2.18) pienempien klustereiden poistamiseksi vähentääkseen klusterien lukumäärää. Tämän jälkeen käytetään kriteeriä (2.17) löytääkseen klusterointi, joka minimoi parhaiten kriteerin. Satunnaisten alkuolosuhteiden käyttämisellä EM-algoritmille on kuitenkin edelleen alustusongelma, joka muuttuu kevyemmäksi suurempien alkuklustereiden lukumäärän avulla.

### Vahva EM-klusterointialgoritmi

Ensin käsitellään $\alpha_k$ termejä EM-sekoitteen tavoitefunktion (2.10) säätämiseksi [8]. Tiedetään, että suhde $\alpha_k$ voi olla yhden luokan $k$ datapisteen todennäköisyys. Näin ollen voidaan käyttää arvoa $-\ln \alpha_k$ luokan $k$ yhden datapisteen esiintymisen informaationa, ja $-\sum_{k=1}^{c} {}_k \ln \alpha_k$ on informaation keskiarvo, jota kutsutaan yleisesti entropiaksi. Kun $\alpha_k = 1/c, \forall k = 1, 2, \ldots, c$ sanotaan, että informaatiota arvosta $\alpha_k$ ei ole. Tässä vaiheessa entropia saavuttaa maksimiarvonsa. Tämän vuoksi aluksi lisätään termi alkuperäiseen

EM-tavoitefunktioon (EM objective function). Tämän jälkeen käytetään oppimisprosessia $\alpha_k$:n estimoimiseksi minimoimalla entropia, jolloin saadaan eniten tietoa arvosta $\alpha_k$. Arvon $-\sum_{k=1}^{c} \alpha_k \ln \alpha_k$ minimoiminen vastaa arvon $\sum_{k=1}^{c} \alpha_k \ln \alpha_k$ maksimointia, tämän vuoksi käytetään jälkimmäistä arvoa EM-tavoitefunktion sakkoterminä (penalty term). Tämän vuoksi ensimmäinen ehdotetu EM-sekoitteen tavoitteellinen tehtävä on maksimoida

$$J(\alpha, \theta) = \sum_{i=1}^{n} \sum_{k=1}^{c} \hat{z}_{ki} \ln[\alpha_k f(x_i; \theta_k)] + \beta \sum_{i=1}^{n} \sum_{k=1}^{c} \alpha_k \ln \alpha_k, \quad \beta \geq 0. \tag{2.19}$$

Tiedetään, että EM-algoritmi on hyvä tapa parametrien estimointiin kunhan on annettu hyvä lähtötilanne [8]. Yhtälön (2.19) sakkotermiä $\sum_{i=1}^{n} \sum_{k=1}^{c} \alpha_k \ln \alpha_k$ käytetään säätämään EM-algoritmia, jossa se on aina negatiivinen. Näin ollen, olkoon $\beta = 0$ algoritmin suorittamisen lopussa, jotta saadaan alkuperäinen EM-estimaatti. Suhde $\alpha_k$ voidaan johtaa maksimoimalla $J(\alpha, \theta)$ $\alpha_k$:n suhteen, rajoituksella $\sum_{k=1}^{c} \alpha_k = 1$ ja seuraavalla päivitetyllä yhtälöllä:

$$\alpha_k^{(uusi)} = \alpha_k^{EM} + \beta \alpha_k^{(vanha)} (\ln \alpha_k^{(vanha)} - \sum_{s=1}^{c} \alpha_s^{(vanha)} \ln \alpha_s^{(vanha)}), \tag{2.20}$$

jossa $\alpha_k^{EM} = \sum_{i=1}^{n} \hat{z}_{ki}/n$.

Yhtälö (2.20) on tärkeä ehdotetulle vahvalle EM-klusterointialgoritmille [8]. Kyseisessä yhtälössä $\sum_{s=1}^{c} \alpha_s \ln \alpha_s$ on $\ln \alpha_k$:n painotettu keskiarvo painoilla $\alpha_1, \ldots, \alpha_c$. $K$:nnessa sekoitussuhteessa $\alpha_k^{(vanha)}$, jos $\ln \alpha_k^{(vanha)}$ on pienempi kuin painotettu keskiarvo, niin uusi sekoitussuhde $\alpha_k^{(uusi)}$ tulee pienemmäksi kuin vanha $\alpha_k^{(vanha)}$. Eli pienempi suhde pienenee ja suurempi suhde kasvaa seuraavassa iteraatiossa ja näin kilpailu tapahtuu. Jos $\alpha_k \leq 0$ tai $\alpha_k < 1/n$ jollain $1 \leq k \leq c^{(vanha)}$, niitä pidetään laittomina suhteina. Tässä tilanteessa hylätään nuo kluserit ja päivitetään klusterien lukumäärä $c^{(vanha)}$:sta seuraavaan:

$$c^{(uusi)} = c^{(vanha)} - |\alpha_k | \alpha_k < 1/n, \quad k = 1, \ldots, c^{vanha}| \tag{2.21}$$

Lisäksi, rajoitusten $\sum_{k'=1}^{c^{(uusi)}} \alpha_{k'} = 1$ ja $\sum_{k'=1}^{c^{(uusi)}} \hat{z}_{k'i} = 1$ säilyttämiseksi, säädetään parametreja $\alpha_{k'}$ ja $\hat{z}_{k'i}$ seuraavanlaisiksi:

$$\alpha_{k'} = \frac{\alpha_{k'}}{\sum_{s=1}^{c^{(uusi)}} \alpha_s} \tag{2.22}$$

$$\hat{z}_{k'i} = \frac{\hat{z}_{k'i}}{\sum_{s=1}^{c^{(uusi)}} \hat{z}_{si}} \tag{2.23}$$

Kilpailukaava-asetuksissa (competition schema setting) algoritmi voi automaattisesti vähentää klustereiden lukumäärää ja samanaikaisesti saada parametreille estimaatteja [8]. Toisaalta, parametri $\beta$ voi auttaa kontrolloimaan kilpailua. Voimme päätellä, että

$$-e^{-1} \leq \alpha_k \ln \alpha_k < 0 \tag{2.24}$$

Jos $0 < \alpha_k \leq 1, \forall k = 1, 2, \ldots, c$, ja olkoon

$$E = \sum_{s=1}^{c} \alpha_s \ln \alpha_s < 0, \tag{2.25}$$

tällöin

$$\alpha_k E = \alpha_k \sum_{s=1}^{c} \alpha_s \ln \alpha_s < 0. \tag{2.26}$$

Käyttäen yhtälöitä (2.24) ja (2.26), saadaan

$$-e^{-1}\beta < \beta\alpha_k \left( \ln \alpha_k - \sum_{s=1}^{c} \alpha_s \ln \alpha_s \right) < \beta(-\alpha_k E) \tag{2.27}$$

Rajoituksessa $\sum_{k=1}^{c} \alpha_k = 1$, ja ainoastaan kun $\alpha_k < 1/2$, voi olla $(\ln \alpha_k - \sum_{s=1}^{c} \alpha_s \ln \alpha_s) < 0$ [8]. Välttääkseen tilannetta, jossa kaikki $\alpha_k \leq 0$, vasen puoli yhtälöstä (2.27) pitää olla suurempi kuin $-\max \alpha_k | \alpha_k < 1/2, k = 1, 2, \ldots, c$. Saatiin $\beta$:n perusolosuhteeksi:

$$\beta < \max \alpha_k e | \alpha_k < 1/2, k = 1, 2, \ldots, c < e/2. \tag{2.28}$$

Jotta vältytään siltä, että $\beta$ on liian suuri, voidaan käyttää $\beta \in [0, 1]$ [8]. Lisäksi, jos $\alpha^{(uusi)}$:n ja $\alpha^{(vanha)}$:n välinen ero on pieni, tällöin $\beta$:n tulisi olla suuri kilpailun lisäämiseksi. Jos taas näiden ero on suuri, tällöin $\beta$:n tulisi olla pieni ylläpitääkseen vakauden. Täten määritellään päivitetty kaava $\beta$:lle.

$$\beta = \frac{\sum_{k=1}^{c} \exp -\eta n | \alpha_k^{(uusi)} - \alpha_k^{(vanha)} |}{c}, \tag{2.29}$$

jossa $\eta$ voidaan asettaa olemaan $\min \left\{ 1, 0.5^{\lfloor \frac{d}{2} - 1 \rfloor} \right\}$, jossa $\lfloor a \rfloor$ tarkoittaa suurinta kokonaislukua, joka on enintään $a$. Yhdistämällä yhtälö (2.20) ja epäyhtälön (2.27) oikea puoli, vältetään $\alpha_{(1)}^{(uusi)} = \max_{1 \leq k \leq c} \alpha_k^{(uusi)} > 1$, olkoon $\alpha_{(1)}^{EM} = \max_{1 \leq k \leq c} \alpha_k^{EM}, \alpha_{(1)}^{(vanha)} = \max_{1 \leq k \leq c} \alpha_k^{(vanha)}$ ja $E = \sum_{k=1}^{c} \alpha_k^{(vanha)} \ln \alpha_k^{(vanha)}$. Tällöin

$$\beta \leq (1 - \alpha_{(1)}^{EM}) / (-\alpha_{(1)}^{(vanha)} E) \tag{2.30}$$

Tällöin saadaan $\beta$:lle seuraavanlainen kaava:

$$\beta = \min \left\{ \frac{\sum_{k=1}^{c} \exp(-\eta n | \alpha_k^{(uusi)} - \alpha_k^{(vanha)}|)}{c}, \frac{(1 - \alpha_{(1)}^{EM})}{(-\alpha_{(1)}^{(vanha)} E)} \right\} \tag{2.31}$$

Tarkastellaan nyt Gaussin sekoitemallia

$$\begin{aligned} f(x; \alpha, \theta) &= \sum_{k=1}^{c} \alpha_k f(x; \mu_k, \Sigma_k) \\ &= \sum_{k=1}^{c} \alpha_k (2\pi)^{(d/2)} |\Sigma_k|^{-(1/2)} e^{-(1/2)(x-\mu_k)' \Sigma_k^{-1} (x-\mu_k)}. \end{aligned} \tag{2.32}$$

Parametrien $\mu_k$ ja $\Sigma_k$ päivitetyt yhtälöt voidaan johtaa seuraaviksi:

$$\mu_k = \frac{\sum_{i=1}^{n} \hat{z}_{ki} x_i}{\sum_{i=1}^{n} \hat{z}_{ki}} \tag{2.33}$$

$$\Sigma_k = \frac{\sum_{i=1}^{n} \hat{z}_{ki}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{n} \hat{z}_{ki}} \tag{2.34}$$

Koska $\beta$ voi hypätä milloin vain, olkoon $\beta = 0$, kun klusterien lukumäärä $c$ on vakaa [8]. Kun klusterien lukumäärä $c$ on vakaa, se tarkoittaa sitä, että se ei enää laske. Pidetään kaikki datapisteet alkuperäisinä keskiarvoina $\mu_k = x_k$, toisinsanoen $\widetilde{c} = n$, ja käytetään $\alpha_k = 1/\widetilde{c}, \forall k = 1, 2, \ldots, \widetilde{c}$ alkuperäisinä sekoitussuhteina. Valitaan myös toteutettavissa oleva kovarianssimatriisi. Olkoon

$$\begin{aligned} D_k &= sort\{d_{ki}^2 = \|x_i - \mu_k\|^2 : d_{ki}^2 > 0, i \neq k, 1 \leq i \leq n\} \\ &= \{d_{k(1)}^2, d_{k(2)}^2, \ldots, d_{k(n')}^2\} \end{aligned} \tag{2.35}$$

ja

$$\Sigma_k = d^2_{k(\lceil \sqrt{\widetilde{c}} \rceil)} \mathbf{I}_d, \tag{2.36}$$

jossa $\mathbf{I}_d$ on $d \times d$ yksikkömatriisi.

Kun käytetään suurempaa klusterien lukumäärää Gaussin jakauman EM-algoritmissa, klusterin kovarianssimatriisi, jolla on hyvin pieni suhde $\alpha_k$, voi olla lähellä yksikköä [8]. Jotta tältä ongelmalta vältytään, käytetään rajoittavaa kovarianssimatriisia $\tilde{\Sigma}_k$ seuraavasti:

$$\tilde{\Sigma}_k = (1 - \gamma)\Sigma_k + \gamma Q, \tag{2.37}$$

jossa $\gamma$ on pieni positiivinen luku ja $Q$ on myös diagonaalimatriisi pienillä positiivisilla luvuilla. Ehdotettu vahva EM-klusterointialgoritmi voidaan siten tiivistää seuraavasti:

**Vahva EM-klusterointi algoritmi**

**Ensimmäinen askel:** Päätetään, että $\varepsilon > 0$. Olkoon alkuarvot $\beta^{(0)} = 1, c^{(0)} = n, \alpha_k^{(0)} = 1/n$ ja $\mu^{(0)} = X$.
**Toinen askel:** Lasketaan $\Sigma_k^{(0)}$ käyttäen yhtälöä (2.36).
**Kolmas askel:** Lasketaan $\hat{z}_{ki}^{(0)}$ parametreilla $\alpha_k^{(0)}, \mu_k^{(0)}$ ja $\Sigma_k^{(0)}$ käyttäen yhtälöä (2.11) ja asetetaan $t = 1$.
**Neljäs askel:** Lasketaan $\mu_k^{(t)}$ parametreilla $\hat{z}_{k1}^{(t-1)}, \ldots, \hat{z}_{kn}^{(t-1)}$ käyttäen yhtälöä (2.33).
**Viides askel:** Päivitetään $\alpha_k^{(t)}$ parametreilla $\hat{z}_{k1}^{(t-1)}, \ldots, \hat{z}_{kn}^{(t-1)}$ käyttäen yhtälöä (2.20).
**Kuudes askel:** Lasketaan $\beta^{(t)}$ parametreilla $\alpha^{(t)}$ ja $\alpha^{(t-1)}$ käyttäen yhtälöä (2.31).
**Seitsemäs askel:** Päivitetään $c^{(t-1)}$ muotoon $c^{(t)}$ hylkäämällä klusterit, joissa $\alpha_k^{(t)} \leq 1/n$ ja säädetään $\alpha_k^{(t)}$ ja $\hat{z}_{ki}^{(t-1)}$ käyttäen yhtälöitä (2.22) ja (2.23). Jos $t \geq 60$ ja $c^{(t-60)} - c^{(t)} = 0$, niin olkoon $\beta^{(t)} = 0$.

**Kahdeksas askel:** Päivitetään $\Sigma_k^{(t)}$ parametreilla $\mu_k^{(t)}$ ja $\hat{z}_{k1}^{(t-1)}, \ldots, \hat{z}_{kn}^{(t-1)}$ käyttäen yhtälöitä (2.34) ja (2.37).
**Yhdeksäs askel:** Päivitetään $\hat{z}_{ki}^{(t)}$ parametreilla $\alpha_k^{(t)}, \mu_k^{(t)}$ ja $\Sigma_k^{(t)}$ käyttäen yhtälöä (2.11)
**Kymmenes askel:** Päivitetään $\mu_k^{(t+1)}$ parametreilla $\hat{z}_{k1}^{(t)}, \ldots, \hat{z}_{kn}^{(t)}$ käyttäen yhtälöä (2.33).
**Yhdestoista askel:** Verrataan parametreja $\mu^{(t+1)}$ ja $\mu^{(t)}$. Jos $\max_{1 \leq k \leq c^{(t)}} \|\mu_k^{(t+1)} - \mu_k^{(t)}\| < \varepsilon$, niin lopetetaan. Ja jos $t = t + 1$ palataan viidenteen askeleeseen.

Algoritmin ensimmäisessä askeleessa kaikki datapisteet on määritetty alkuperäisiksi arvoiksi $c^{(0)} = n, \alpha_k^{(0)} = 1/n$ ja $\mu^{(0)} = X$ [8]. Itseasiassa algoritmi voi kulkea samalla tavalla kuin Figueiredon ja Jainin menetelmä, joka käyttää joitain satunnaisia alkuarvoja syöttämällä suurempia klustereiden lukumääriä, määrittämättä kaikkia datapisteitä alkuperäisiksi arvoiksi. Tässä tapauksessa voidaan satunnaisesti valita $\widetilde{c}$ datapistettä aineistosta niin, että $\widetilde{c}$ on pienempi kuin $n$ alkuperäisillä keskiarvoilla $\mu_1^{(0)}, \mu_2^{(0)}, \ldots, \mu_{\widetilde{c}}^{(0)}$. Tällöin alkuperäisistä suhteista tulee $\alpha_k^{(0)} = 1/\widetilde{c}$ $k$:n arvoilla $1, 2, \ldots, \widetilde{c}$. Muut algoritmin askeleet pysyvät muuttumattomina.

## 2.5 Pääkomponenttianalyysi

Pääkomponenttianalyysi (PCA) on prosessi, jolla lasketaan pääkomponentit ja näiden komponenttien myöhempi käyttö aineiston ymmärtämiseen [2]. PCA on ohjaamaton lähestymistapa, sillä se sisältää ainoastaan joukon ominaisuuksia $X_1, X_2, \ldots, X_p$, eikä liittyvää vastausta $Y$. PCA myös tarjoaa työkalun aineiston visualisointiin.

Oletetaan, että halutaan visualisoida $n$ havainnot mittauksilla joukon $p$ ominaisuuksille $X_1, X_2, \ldots, X_p$, osana tutkivaa data-analyysia [2]. Tämä voidaan tehdä tarkastelemalla aineiston kaksiuloitteisia hajontakuvioita, joista jokainen sisältää $n$ havainnon mittauksia kahdesta ominaisuudesta. Tällaisia hajontakuvioita on olemassa $\binom{p}{2} = p(p-1)/2$ kappaletta. Jos $p$ on suuri, kaikkia näitä ei pysty tutkimaan. Tämän lisäksi on todennäköistä, että mikään näistä ei ole informatiivinen, sillä ne sisältävät vain pienen osan aineiston kokonaistiedosta. Tällaisessa tilanteessa on oltava parempi menetelmä $n$ havainnon visualisointiin. Erityisesti olisi löydettävä aineiston matalauloitteinen esitys, joka vangitsee mahdollisimman suuren osan tiedoista. Esimerkiksi, jos on mahdollista saada aineistosta kaksiulotteinen esitys, joka vangitsee suurimman osan informaatiosta, niin havainnot voidaan piirtää tähän pienen ulottuvuuden avaruuteen.

PCA tarjoaa työkalun tämän tekemiseen [2]. Sen avulla löydetään aineistosta matalaulotteinen esitys, joka sisältää mahdollisimman paljon vaihtelua. Ajatuksena tässä on se, että jokainen $n$ havainnoista elää $p$-ulotteisessa avaruudessa, kaikki näistä ulottuvuuksista eivät ole kuitenkaan yhtä mielenkiintoisia. PCA hakee pienen määrän mahdollisimman mielenkiintoisia ulottuvuuksia, joissa käsite kiinnostavuudesta mitataan kunkin ulottuvuuden mukaan vaihtelevien havaintojen määrällä. Jokainen PCA:n löytämä ulottuvuus

on ominaisuuksien $p$ lineaarikombinaatio.

## 2.6 Suhteellinen muutos

Suhteellinen muutos on ratkaisevassa asemassa muun muassa laskettaessa indeksinumeroita, ja tuottavuuden mittaamisessa [6]. Sillä on myös mahdollisia merkittäviä etuja taloudellisten suhteiden arviomisessa muuttujien tasoon nähden. Suhteellinen muutos lasketaan seuraavalla kaavalla

$$100 \times \frac{|y - x|}{|x|}, \tag{2.38}$$

jossa $y$ on arvioitu arvo ja $x$ on todellinen arvo [1].

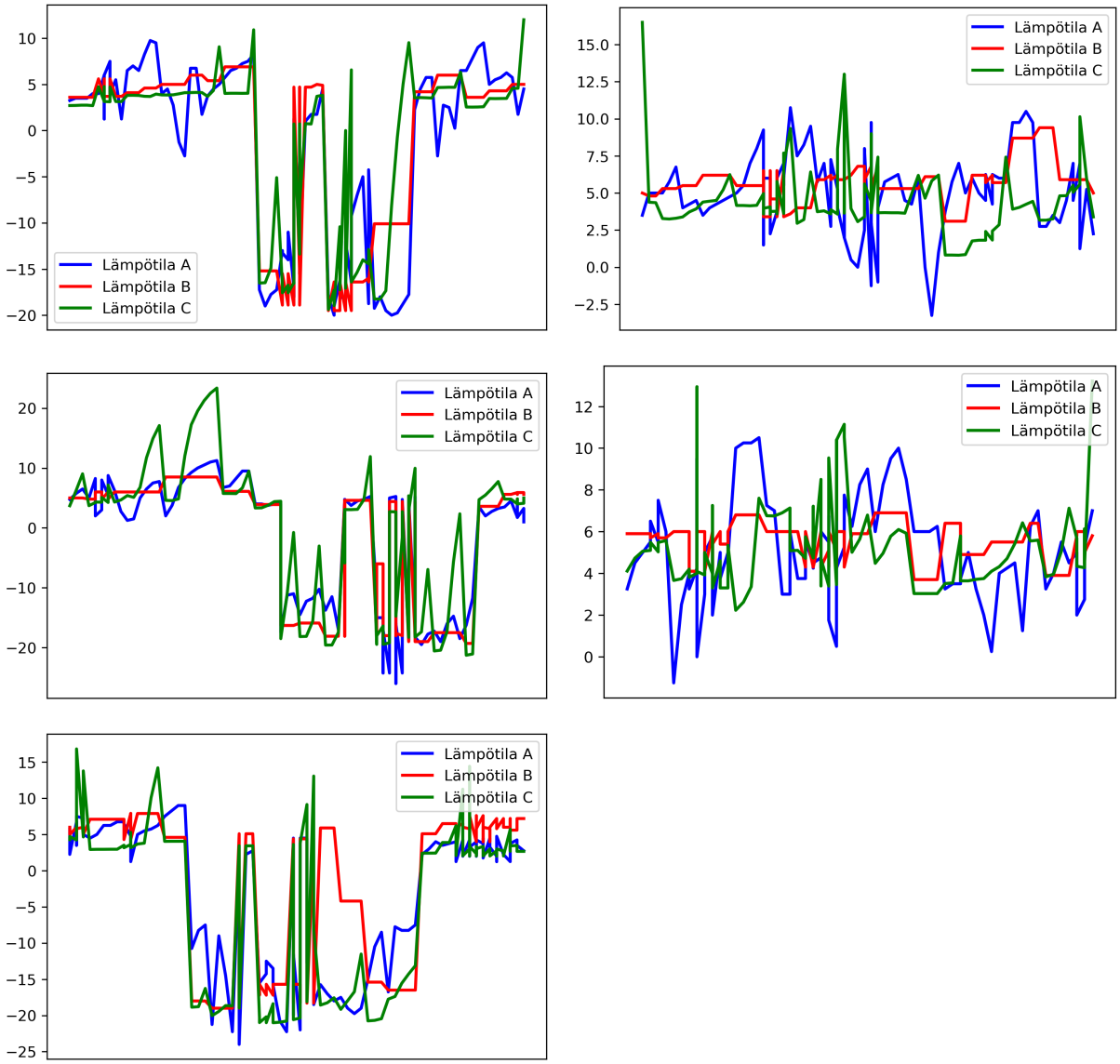
# 3. Tulokset

Tutkielman aineisto koostuu 467 mittauksesta, jotka on suoritettu aikavälillä kesäkuu 2019 - marraskuu 2019 ympäri Suomea. Aineisto koostuu asiakkaan tiedoista, tuotetiedoista, kuljetustiedoista sekä lämpötiloista. Tutkimuksen kannalta mielenkiintoisia muuttujia ovat lämpötilat A, B ja C, joten alkuperäisestä aineistosta poistetaan mittaukset, joissa ei ole tietoa näistä lämpötiloista.

Lopullinen aineisto karsintojen jälkeen on 1105-rivinen taulukko, joka koostuu 192 erillisestä mittauksesta ja se on jaettu kymmeneen yhtä suureen osaan, jotta tuloksista saadaan selkeämpiä ja niitä on helpompi analysoida.

Tulokset koostuvat kuvaajista, jotka kertovat lämpötilojen muutokset, ja klusteroinneista $k$-means sekä EM-GMM menetelmillä. EM-GMM menetelmä on toteutettu kahdella eri tavalla, toisessa menetelmässä sovelletaan PCA:ta klusterointimenetelmään. Tämän lisäksi vastataan tutkimuskysymykseen käyttäen suhteellista muutosta.

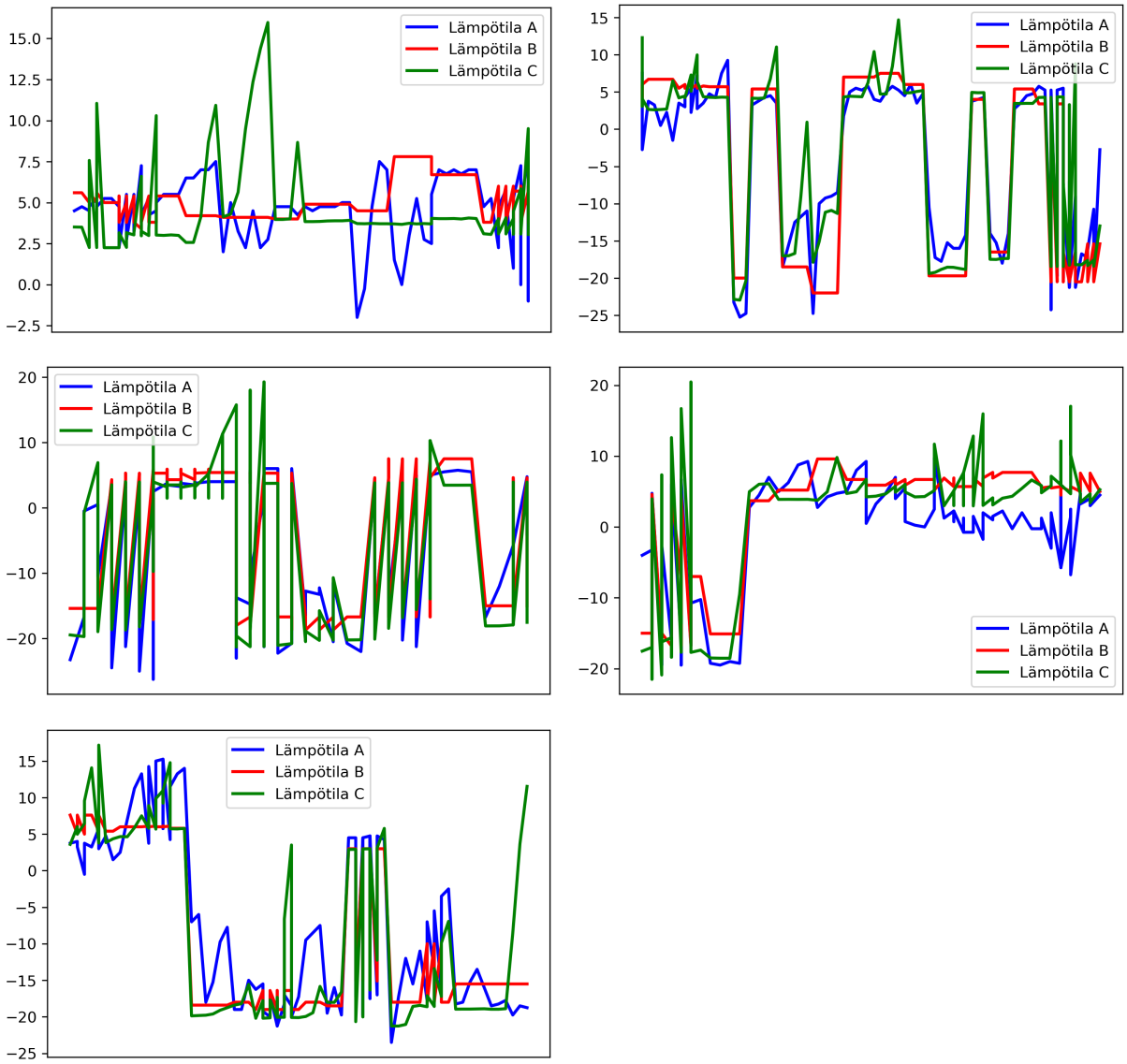
## 3.1 Lämpötilojen vaihtelut




**Kuva 3.1:** Lämpötilat viidessä ensimmäisessä aineiston osajoukossa.

Kuvista 3.1 ja 3.2 nähdään mitattujen lämpötilojen vaihtelut mittauskerroittain aikajärjestyksessä. Kuvaajissa x-akselilla on aika ja y-akselilla lämpötila. Sininen viiva kuvaa lämpötilaa A, punainen viiva lämpötilaa B ja vihreä viiva kuvaa lämpötilaa C.

Kuvista huomataan, että lämpötilojen vaihtelut ovat melko tasaisia. Ajoittain huomataan kuitenkin suuria hyppäyksiä lämpötilassa C (vihreä viiva), ja nämä voidaan selittää muun muassa sillä, että mittauslaite on tippunut tuotteesta, sillä näissä tilanteissa usein huomataan, että lämpötila B (punainen viiva) kuitenkin pysyy lähellä lämpötilaa A. Myös lämpötilassa B näkyy loikkaus kuvan 3.1 viimeisessä kuvaajassa, tämä voidaan selittää mahdollisesti sillä, että lämpötilan mittauksessa on ollut viivettä toimitusasiak-



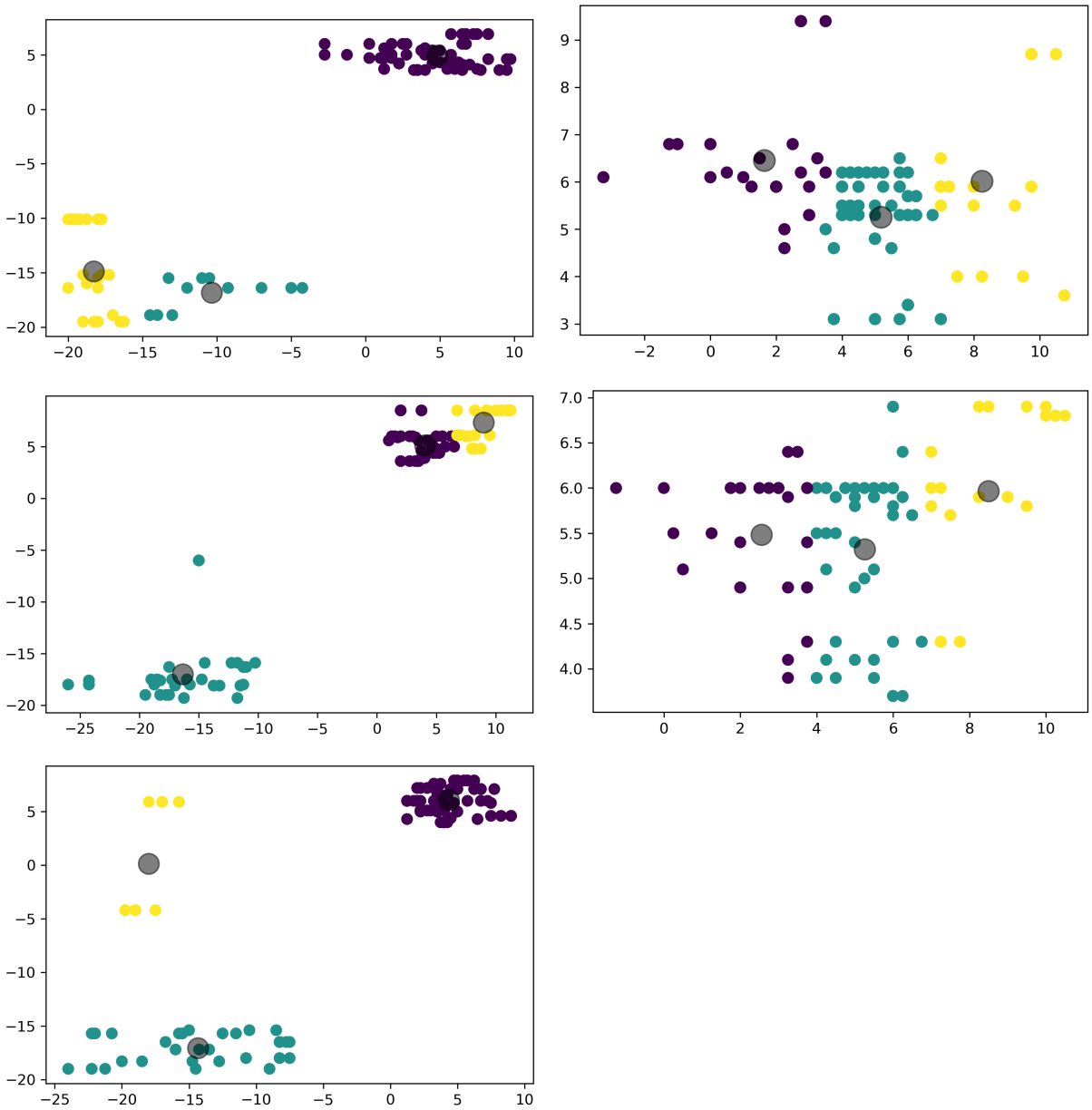

**Kuva 3.2:** Lämpötilat viidessä viimeisessä aineiston osajoukossa.

kaalla, jonka vuoksi tarkkaa lämpötilaa ei saatu. Nopeat lämpötilan vaihtelut voidaan selittää myös kuljetusauton ovien avaamisena tai mittauskerran muutoksena.

## 3.2 $K$-means -klusterointi

Kuvissa 3.3 ja 3.4 on kuvattu jokaisen aineiston osajoukon jakaminen kolmeen eri klusteriin käyttäen $k$-means -klusterointia. Kuvaajissa x-akseli kuvaa lämpötilaa A ja y-akseli kuvaa lämpötilaa B. Jokaisessa kuvaajassa näkyy kolme klusterikeskusta, jotka on merkitty harmaalla ympyrällä.

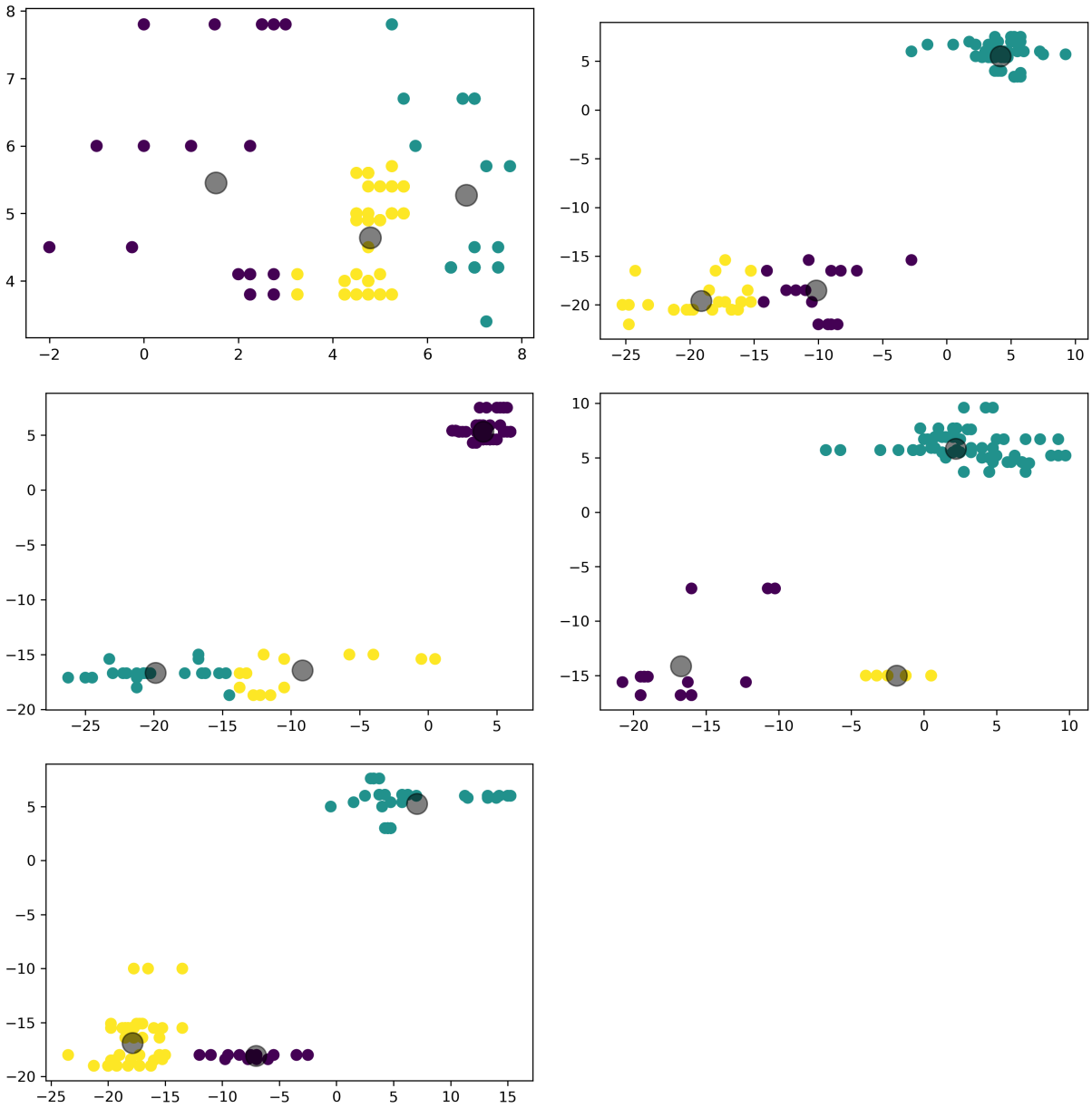
Suurimmassa osassa kuvaajista näkee selkeät erilliset klusterit, mutta osassa klusterit ovat hieman epäselvät eikä niin yhtenäiset. Esimerkiksi kuvan 3.3 toinen ja neljäs klus-



**Kuva 3.3:** $K$-means -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa B.

terointi sekä kuvan 3.4 ensimmäinen klusterointi kuvaavat epäselviä klustereita, voidaan päätellä, että näissä mittauksissa on ollut joitakin ongelmia. On esimerkiksi mahdollista, että näissä mittauksissa kuljetusauton ovea on avattu useammin, eli pudotuksia on ollut enemmän tai esimerkiksi lämpötilaa B ei olla mitattu heti, jolloin ei olla saatu tarkkaa tulosta. Muissa klusteroinneissa näkee selkeästi tuoteryhmät A ja B erikseen ja voidaan päätellä, että mittaukset ovat olleet onnistuneet näissä osajoukoissa.

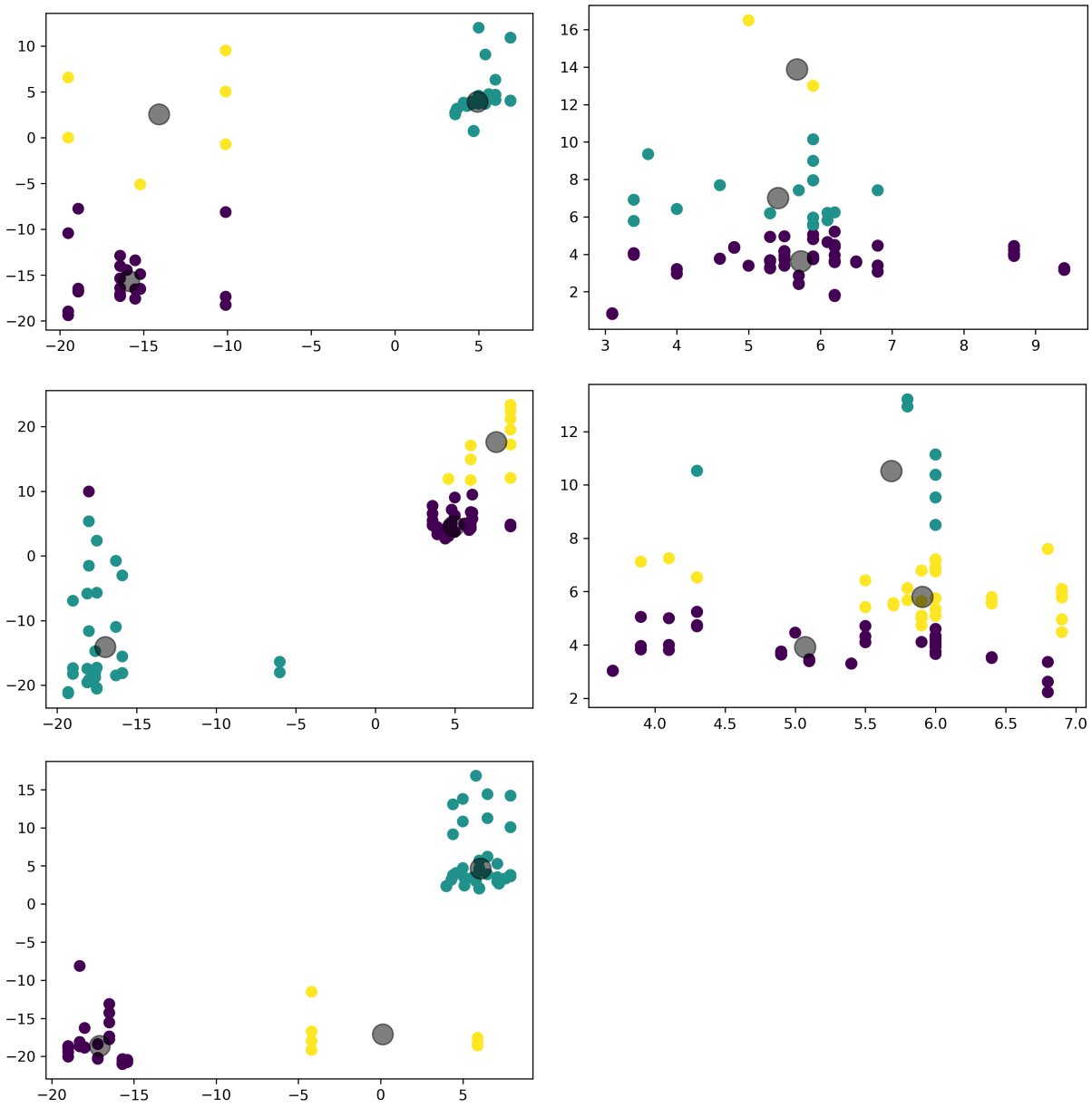
Kuvissa 3.5 ja 3.6 on kuvattu jokaisen aineiston osajoukon jakaminen kolmeen eri klusteriin käyttäen $k$-means -klusterointia. Kuvaajissa x-akseli kuvaa lämpötilaa B ja y-akseli kuvaa lämpötilaa C. Jokaisessa kuvaajassa näkyy kolme klusterikeskusta, jotka on



**Kuva 3.4:** $K$-means -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa B.

merkitty harmaalla ympyrällä.

Kuvista löytyy selkeitä klustereita, mutta ei niin selkeitä tai yhtä paljon kuin kuvissa 3.3 ja 3.4. Kuvan 3.5 ensimmäisessä ja viidennessä kuvaajassa sekä kuvan 3.6 toisessa, neljännessä ja viidennessä kuvaajassa on selkeät klusterit. Näiden lisäksi kuvan 3.5 kolmannessa kuvaajassa ja kuvan 3.6 kolmannessa kuvaajassa klusterit ovat muuten melko selkeät, mutta molemmissa on yksi hajapiste, jonka silmämääräisesti pitäisi kuulua toiseen klusteriin, tämä voidaan selittää mittausvirheellä. Muiden kuvaajien klusterit ovat melko epäselviä ja tämän voi selittää muun muassa sillä, että osissa mittauksia lämpötilat C ovat virheellisesti mitattuja, sillä mittarit saattavat pudota helposti tuotteesta matkan

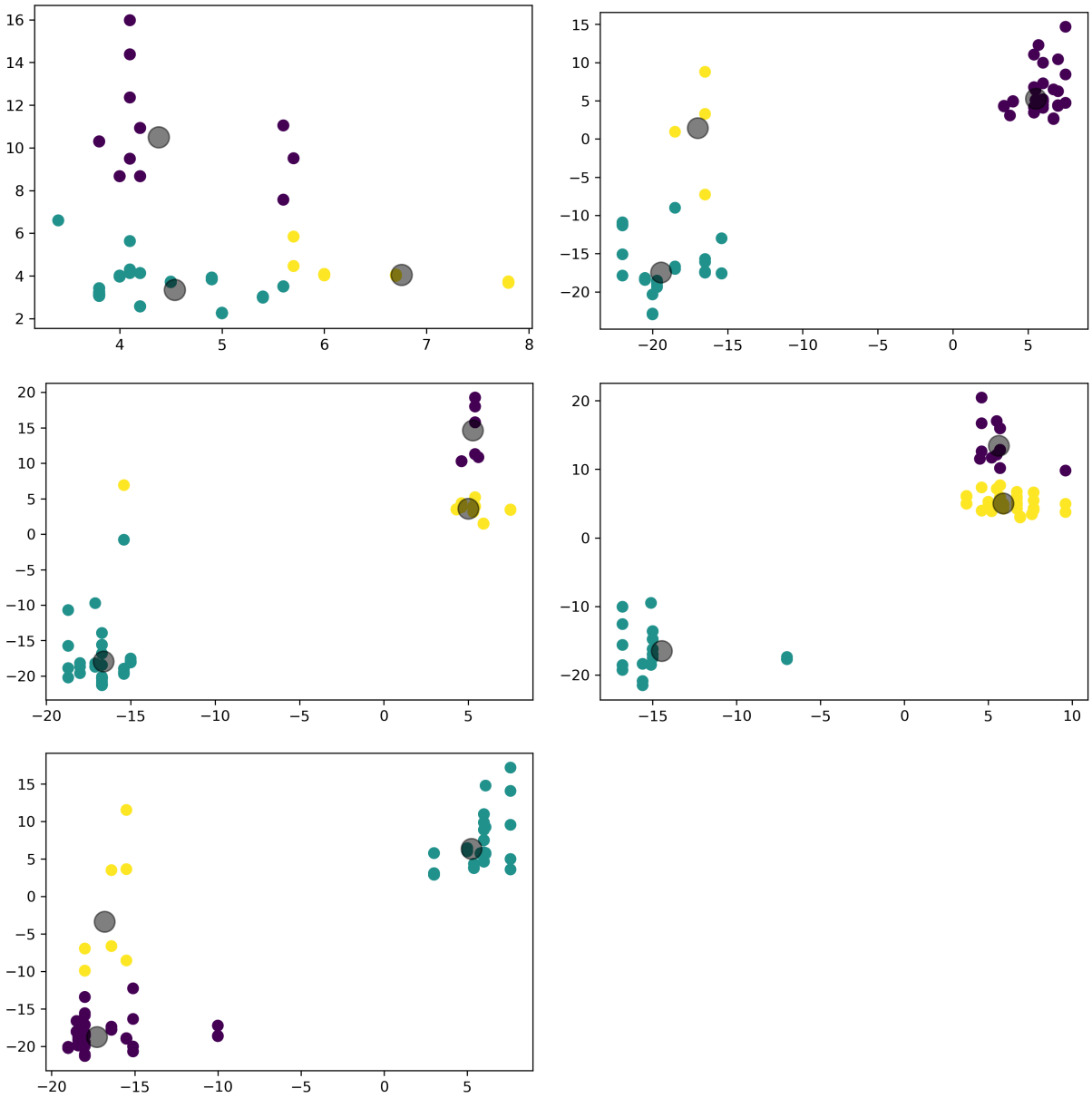


**Kuva 3.5:** $K$-means -klusterointi, jossa x-akseli kuvaa lämpötilaa B ja y-akseli lämpötilaa C.

varrella tai toimitusasiakkaalla, jolloin lämpötila on tietenkin virheellinen.

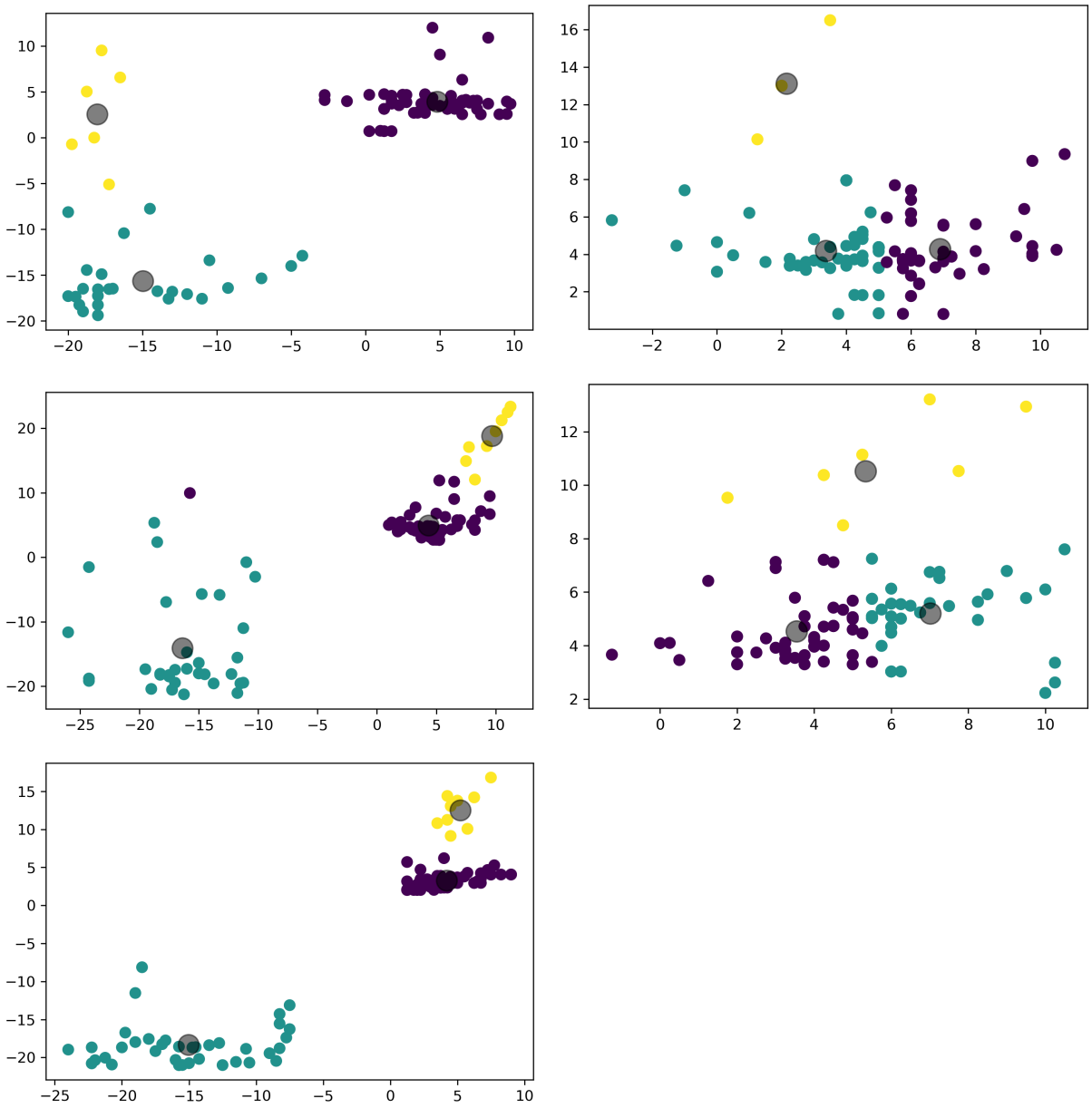
Kuvissa 3.7 ja 3.8 on kuvattu jokaisen aineiston osajoukon jakaminen kolmeen eri klusteriin käyttäen $k$-means -klusterointia. Kuvaajissa x-akseli kuvaa lämpötilaa A ja y-akseli kuvaa lämpötilaa C. Jokaisessa kuvaajassa näkyy kolme klusterikeskusta, jotka on merkitty harmaalla ympyrällä.

Kuvan 3.7 toisesta ja viidennestä kuvaajasta ja kuvan 3.8 toisesta, kolmannesta ja neljännestä kuvaajasta löytyy selkeät klusterit, joissa on huomattavissa erikseen tuoteryhmät A ja B. Kuvassa 3.7 kolmas kuvaaja on muuten hyvin selkeä, mutta sieltä löytyy yksi mittaus, joka silmämääräisesti kuuluisi eri klusteriin kun mihin se on klusteroitu, tämän voi perustella mittausvirheellä. Kuvan 3.7 toisessa ja neljännessä

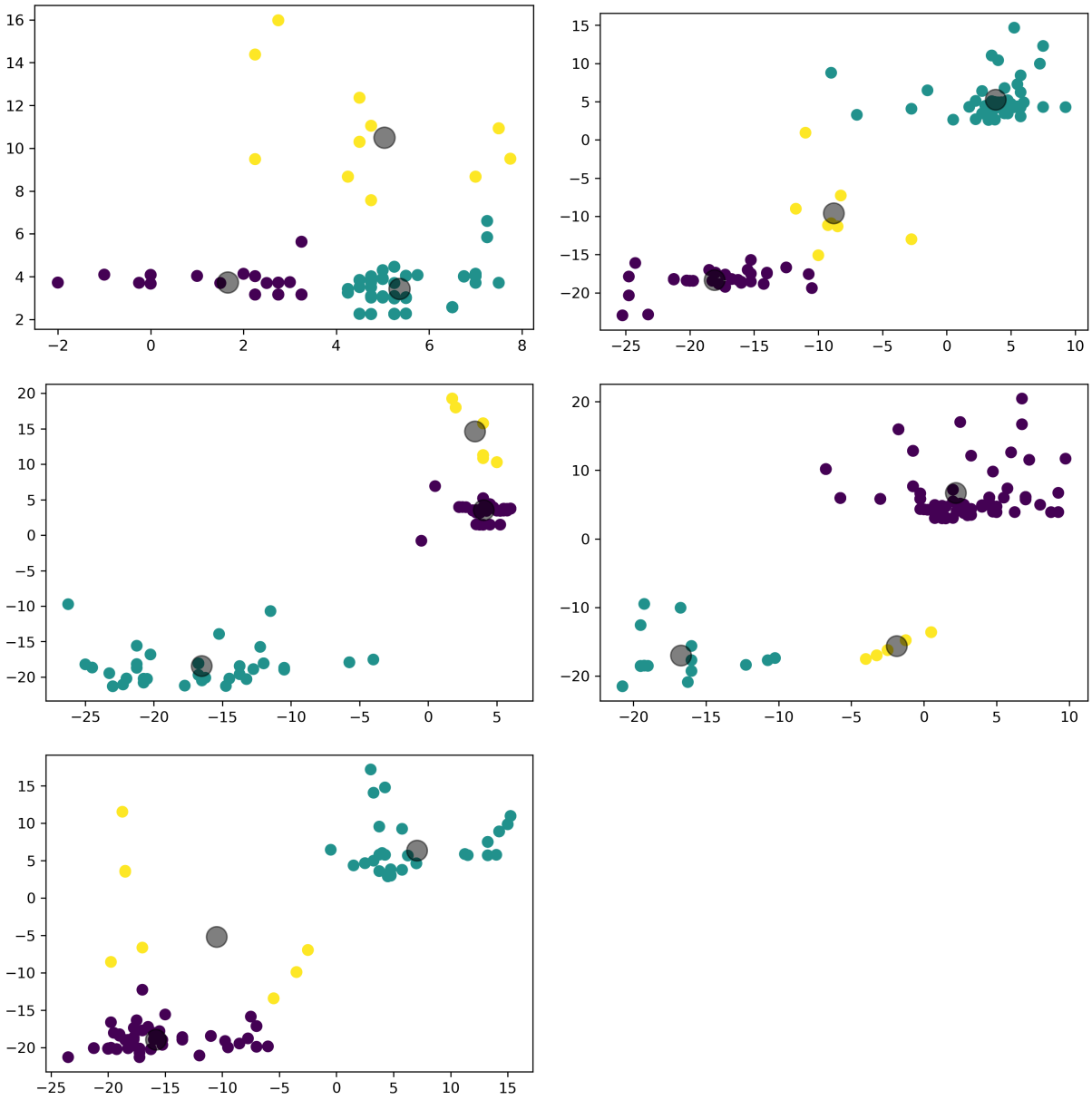


**Kuva 3.6:** $K$-means -klusterointi, jossa x-akseli kuvaa lämpötilaa B ja y-akseli lämpötilaa C.

kuvaajassa klusterit ovat laajalle levinneitä ja ei niin selkeitä ja tämän voisi perustella suurella lämpötilan vaihtelulla, joka saattaa johtua muun muassa kuljetusauton ovien avaamisesta ja mittausvirheistä.




**Kuva 3.7:** $K$-means -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa C.

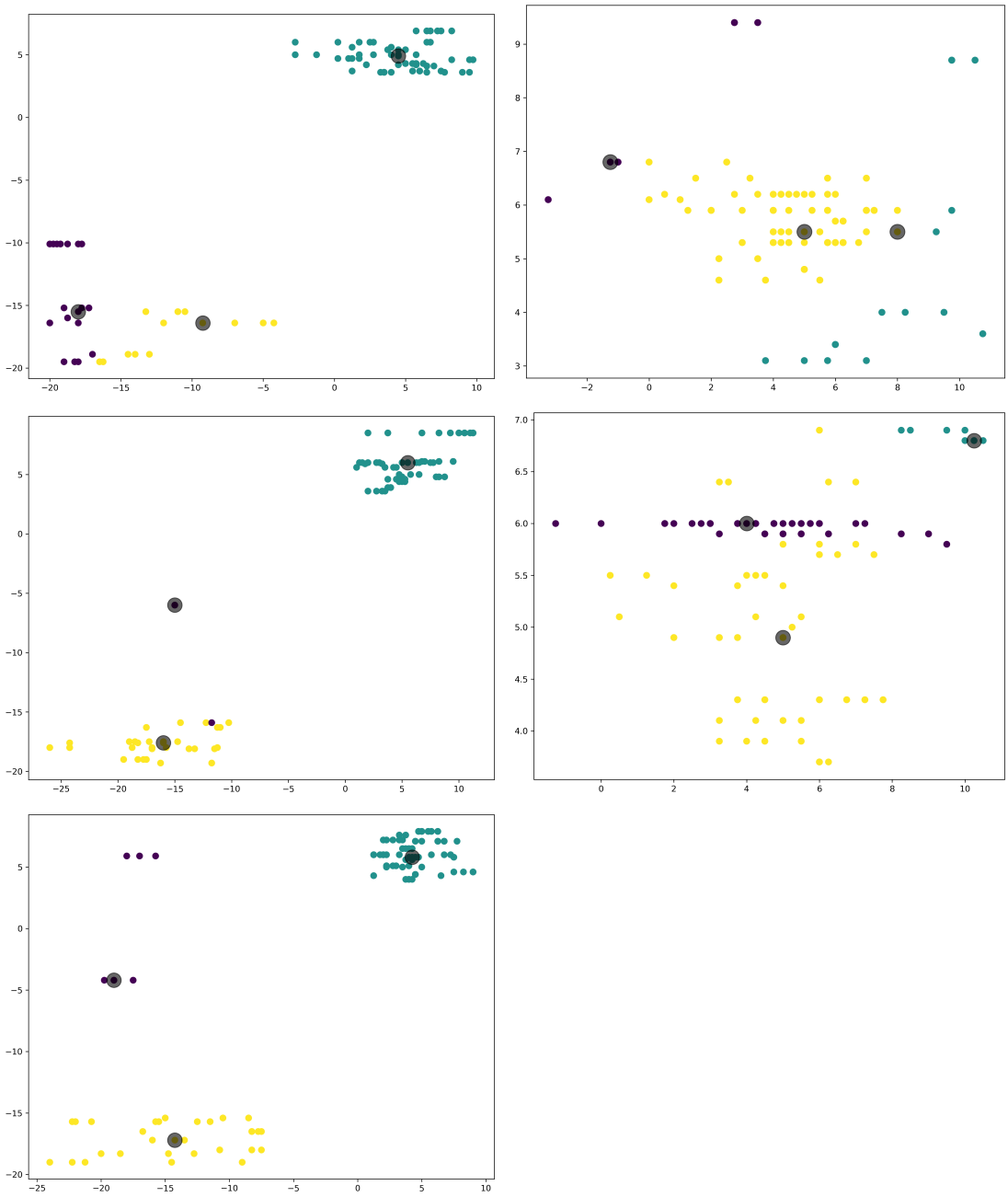


**Kuva 3.8:** $K$-means -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa C.

## 3.3 EM-GMM -klusterointi

Kuvissa 3.9 ja 3.10 on kuvattu jokaisen aineiston osajoukon jakaminen kolmeen eri klusteriin sovittaen EM-algoritmia Gaussin sekoitemallille. Kuvaajissa x-akseli kuvaa lämpötilaa A ja y-akseli kuvaa lämpötilaa B. Jokaisessa kuvaajassa näkyy kolme klusterikeskusta, jotka on merkitty harmaalla ympyrällä.

Kuvan 3.9 viimeisessä kuvaajassa ja kuvan 3.10 neljännessä kuvaajassa klusteroinnit ovat tismalleen samat kuin aikaisemmassa $k$-means klusteroinnissa, tämän lisäksi kuvan 3.9 ensimmäisessä kuvaajassa ja kuvan 3.10 ensimmäisessä, toisessa, kolmannessa ja viidennessä kuvaajassa klusteroinnit ovat melko samat kuin $k$-means klusteroinnissa. Mutta

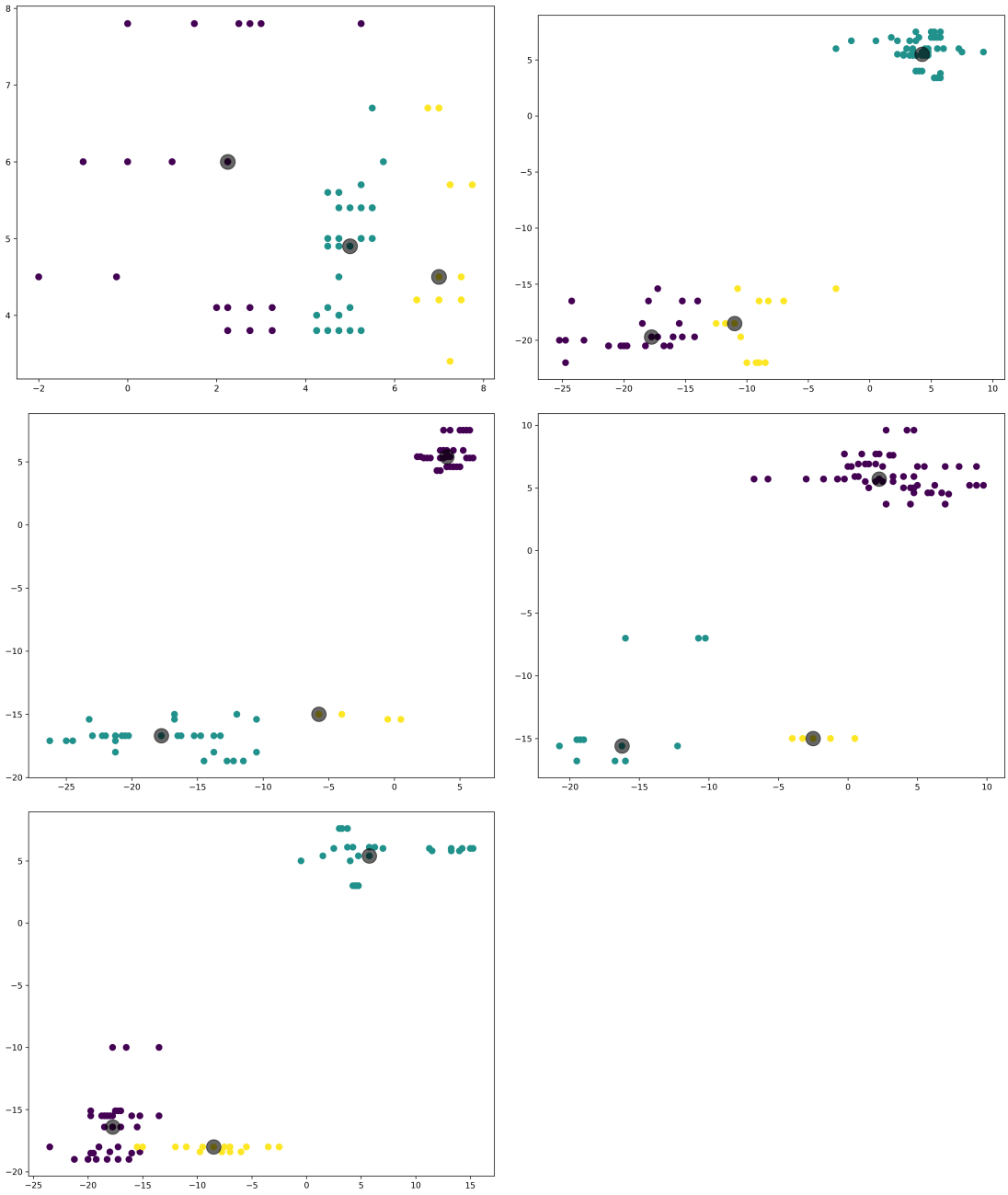


**Kuva 3.9:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa B. Ilman pääkomponenttianalyysiä.

täysin erilaiset klusteroinnit ovat kuvan 3.9 toisessa, kolmannessa ja neljännessä kuvaajassa.

Kuvan 3.9 toisessa ja neljännessä kuvaajassa klusteroinnit ovat vielä epäselvemmät kun samaisten osajoukkojen $k$-means klusteroinneissa ja sekin voidaan selittää muun muassa mittausvirheillä. Kuvan 3.10 ensimmäisessä, toisessa ja kolmannessa kuvaajassa klusterit ovat selkeämmin erilliset kuin vastaavissa $k$-means klustereissa.

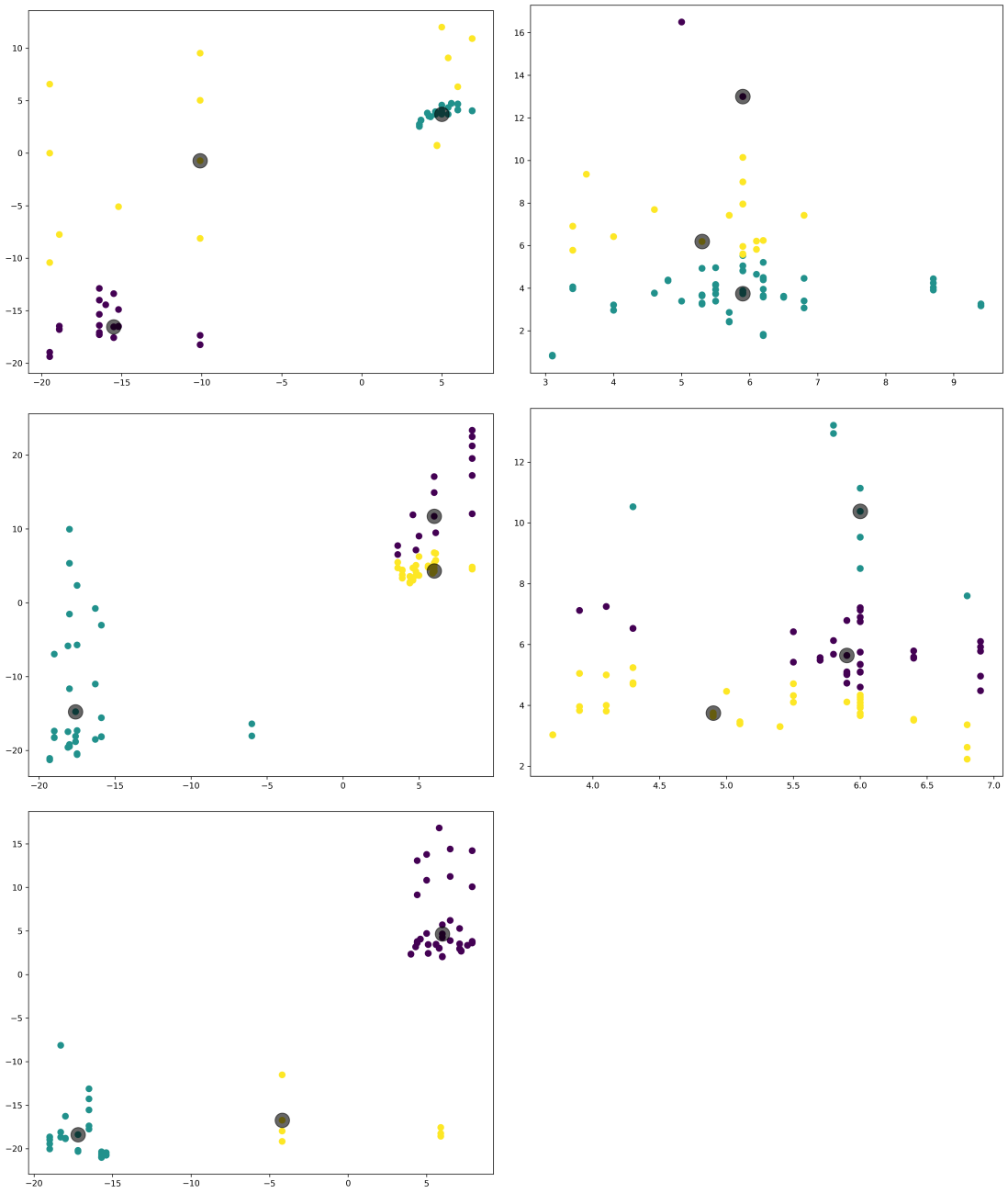
Kuvissa 3.11 ja 3.12 on kuvattu jokaisen aineiston osajoukon jakaminen kolmeen



**Kuva 3.10:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa B. Ilman pääkomponenttianalyysiä.

eri klusteriin sovittaen EM-algoritmia Gaussin sekoitemallille. Kuvaajissa x-akseli kuvaa lämpötilaa B ja y-akseli kuvaa lämpötilaa C. Jokaisessa kuvaajassa näkyy kolme klusterikeskusta, jotka on merkitty harmaalla ympyrällä.

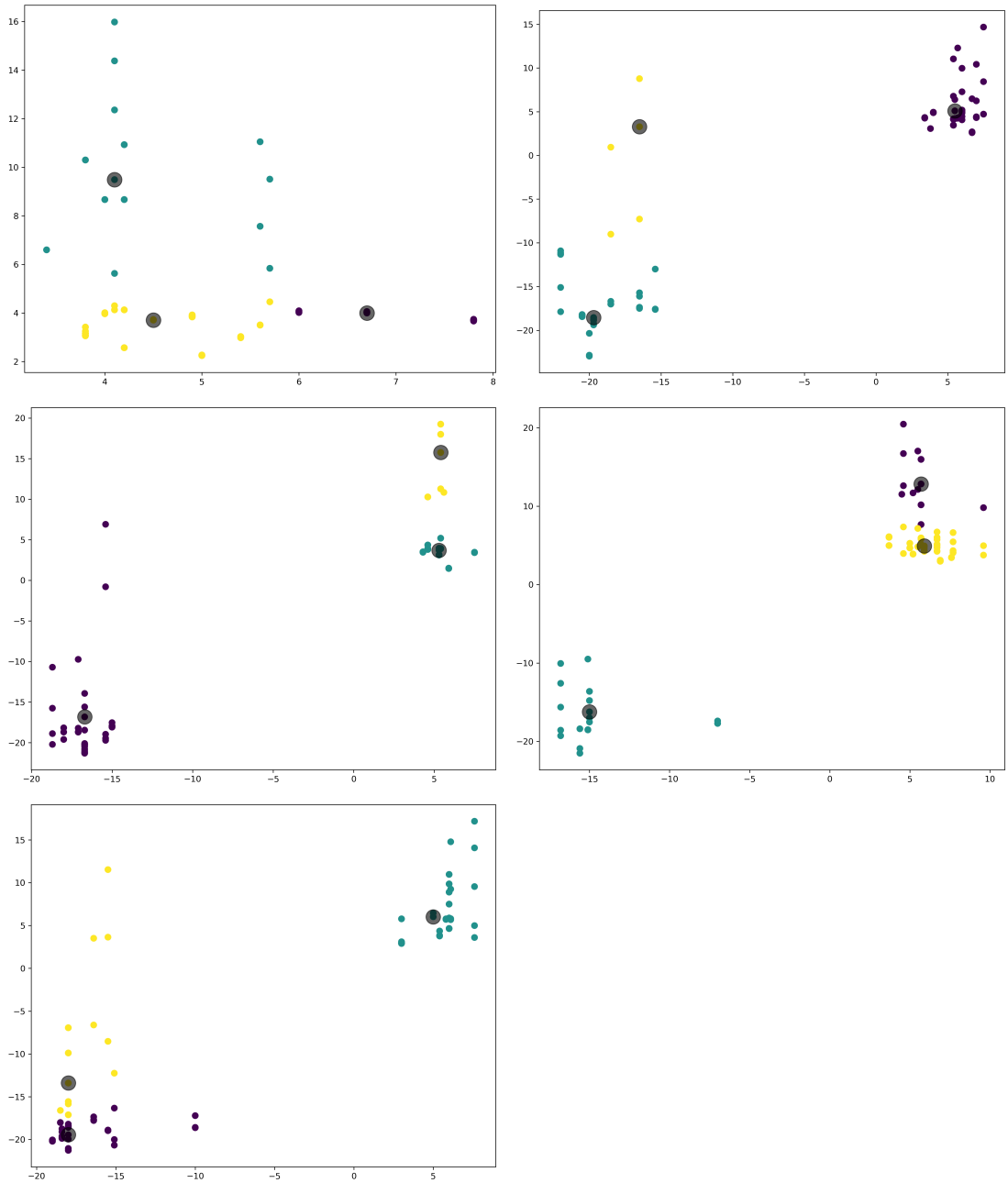
Kuvan 3.11 toisessa ja viidennessä kuvaajassa klusteroinnit ovat tismalleen samat kuin aikaisemmassa $k$-means klusteroinnissa, tämän lisäksi kuvan 3.11 kolmannessa ja neljännessä kuvaajassa sekä kuvan 3.12 kaikissa kuvaajissa klusteroinnit ovat melko samat kuin $k$-means klusteroinnissa. Mutta täysin erilainen klusterointi on kuvan 3.11 ensim-



**Kuva 3.11:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa B ja y-akseli lämpötilaa C. Ilman pääkomponenttianalyysiä.

mäisessä kuvaajassa.

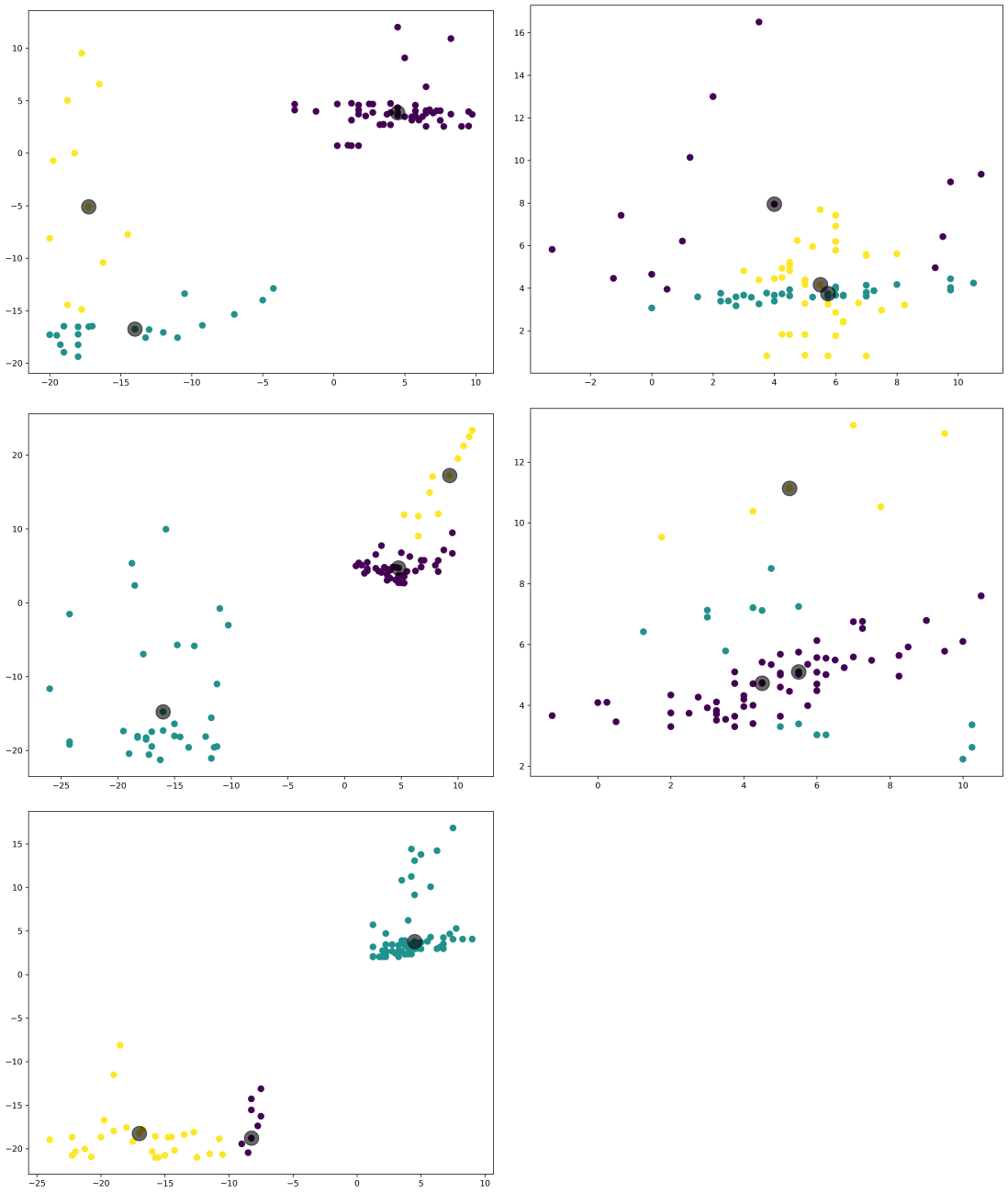
Kuvan 3.11 kolmannessa ja kuvan 3.12 kolmannessa kuvaajassa klusteroinnit ovat selkeämpiä kuin samaisten osajoukkojen $k$-means klusteroinneissa, sillä EM-GMM klusteroinneissa ei ole hajamittauksia, joka voidaan selittää muun muassa mittausvirheillä, tämän lisäksi kuvan 3.12 toisessa kuvaajassa klusterit ovat selkeämmin erillään kuin vastaavassa $k$-means klusteroinnissa. Muissa kuvien 3.11 ja 3.12 klusteroinneissa klusterit ovat hieman epäselvemmin erillään kuin vastaavissa $k$-means klusteroinneissa.



**Kuva 3.12:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa B ja y-akseli lämpötilaa C. Ilman pääkomponenttianalyysiä.

Kuvissa 3.13 ja 3.14 on kuvattu jokaisen aineiston osajoukon jakaminen kolmeen eri klusteriin sovittaen EM-algoritmia Gaussin sekoitemallille. Kuvaajissa x-akseli kuvaa lämpötilaa A ja y-akseli kuvaa lämpötilaa C. Jokaisessa kuvaajassa näkyy kolme klusterikeskusta, jotka on merkitty harmaalla ympyrällä.

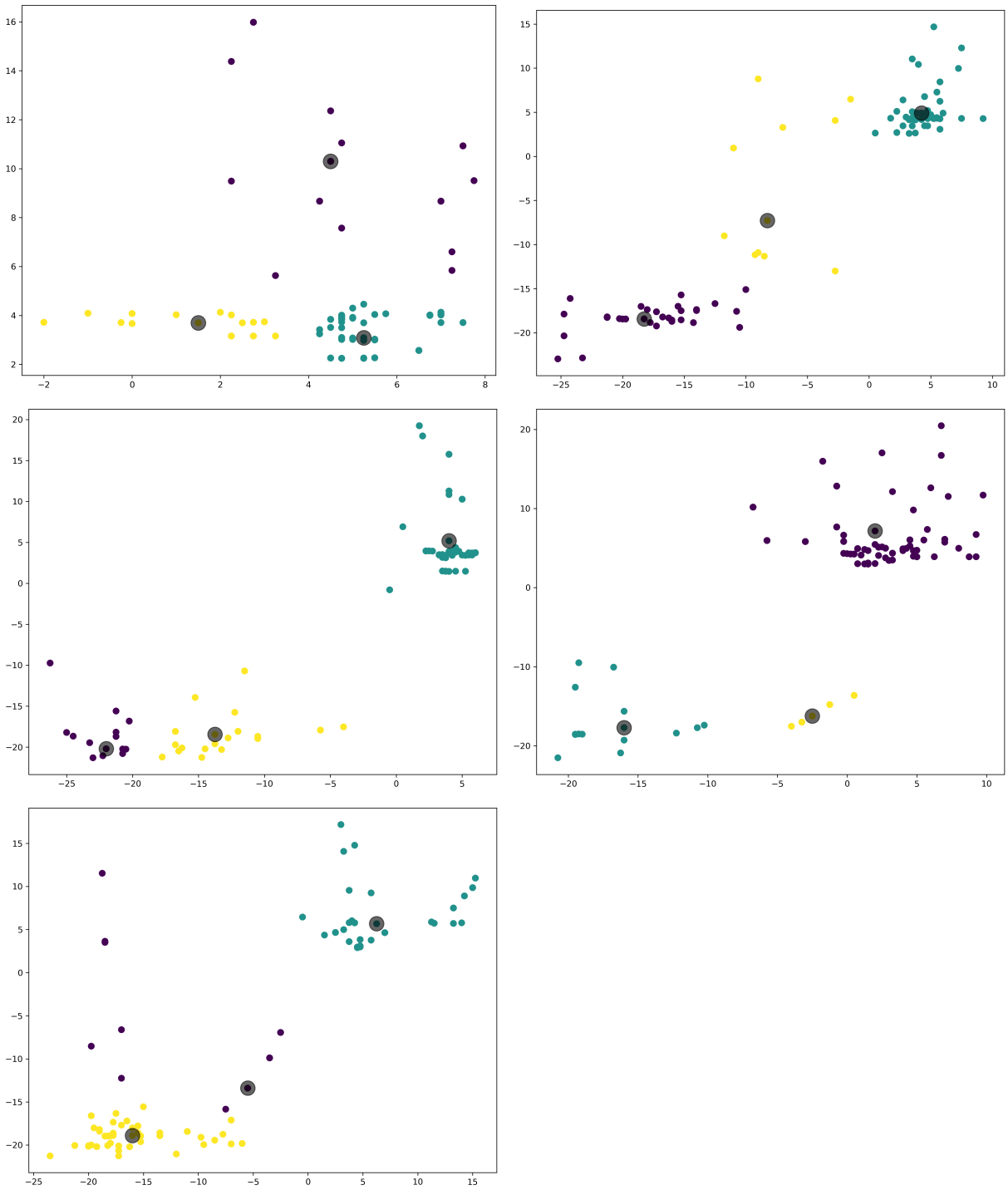
Kuvan 3.14 neljännessä kuvaajassa klusterit ovat tismalleen samat kuin vastaavan osajoukon $k$-means klusteroinnissa, tämän lisäksi kuvan 3.13 ensimmäisessä ja kolmannessa kuvaajassa sekä kuvan 3.14 ensimmäisessä, toisessa ja viidennessä kuvaajassa klus-



**Kuva 3.13:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa C. Ilman pääkomponenttianalyysiä.

teroinnit ovat melko samat kuin vastaavissa $k$-means klusteroinneissa. Täysin erilainen klusterointi on kuvan 3.13 toisessa, neljännessä ja viidennessä sekä kuvan 3.14 kolmannessa kuvaajassa.

Kuvan 3.13 kolmannessa kuvaajassa klusterointi on selkeämpi kuin samaisen osajoukon $k$-means klusteroinnissa, sillä EM-GMM klusteroinnissa ei ole hajamittausta, joka voidaan selittää muun muassa mittausvirheillä, tämän lisäksi kuvan 3.14 kolmannessa kuvaajassa klusterointi on selkeämpi kuin samaisen $k$-means klusteroinnin. Vastaavasti

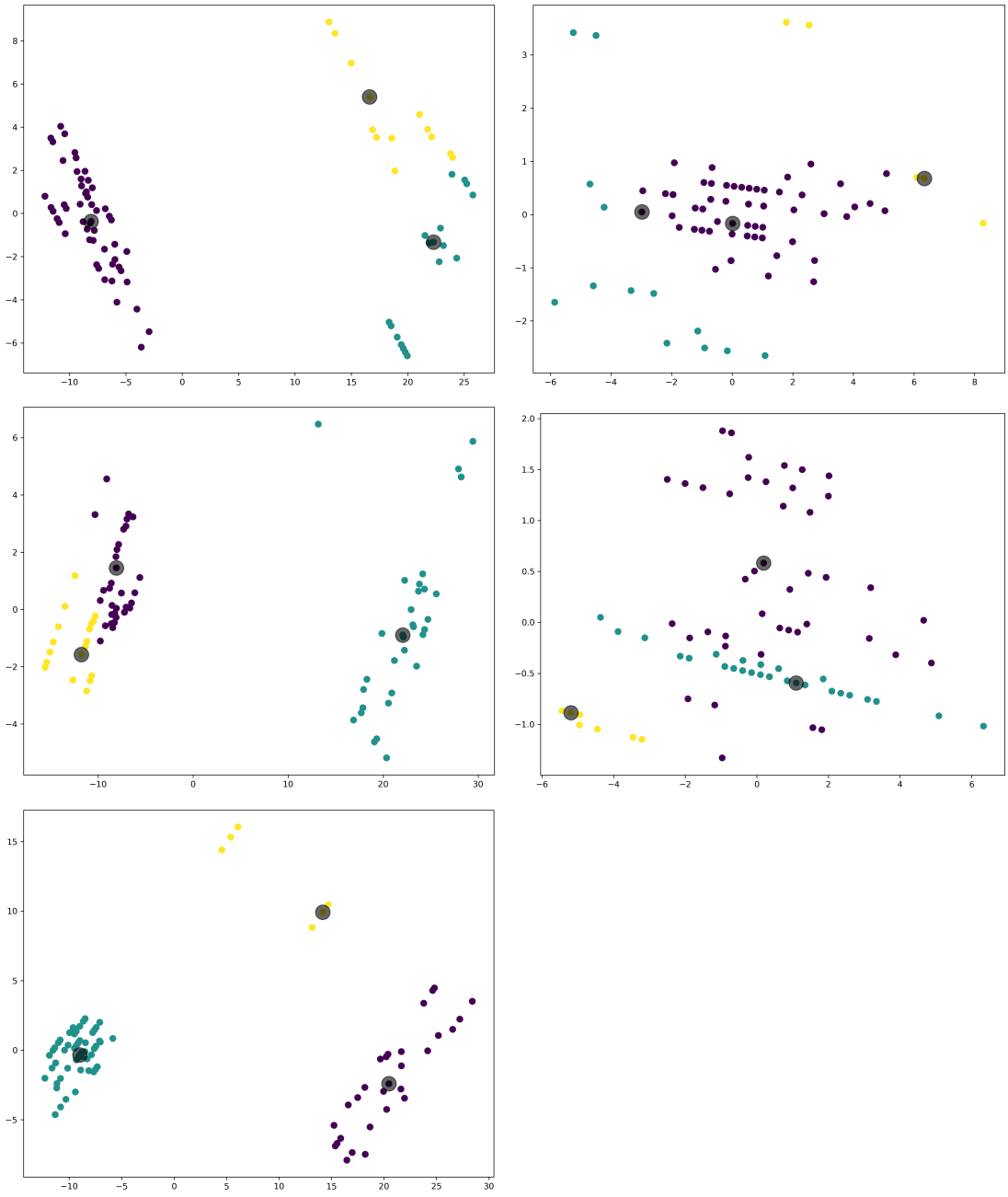


**Kuva 3.14:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa C. Ilman pääkomponenttianalyysiä.

kuvan 3.13 toisessa, neljännessä ja viidennessä sekä kuvan 3.14 ensimmäisessä, toisessa ja viidennessä kuvaajassa klusterit ovat epäselvempiä kuin vastaavassa $k$-means klusteroinnissa. On myös kuvaaja, jossa klusteroinnit ovat erilaiset molemmissa menetelmissä, mutta molemmat klusteroinnit ovat yhtä selkeitä. Tämä tapaus on kuvan 3.13 ensimmäinen

kuvaaja.

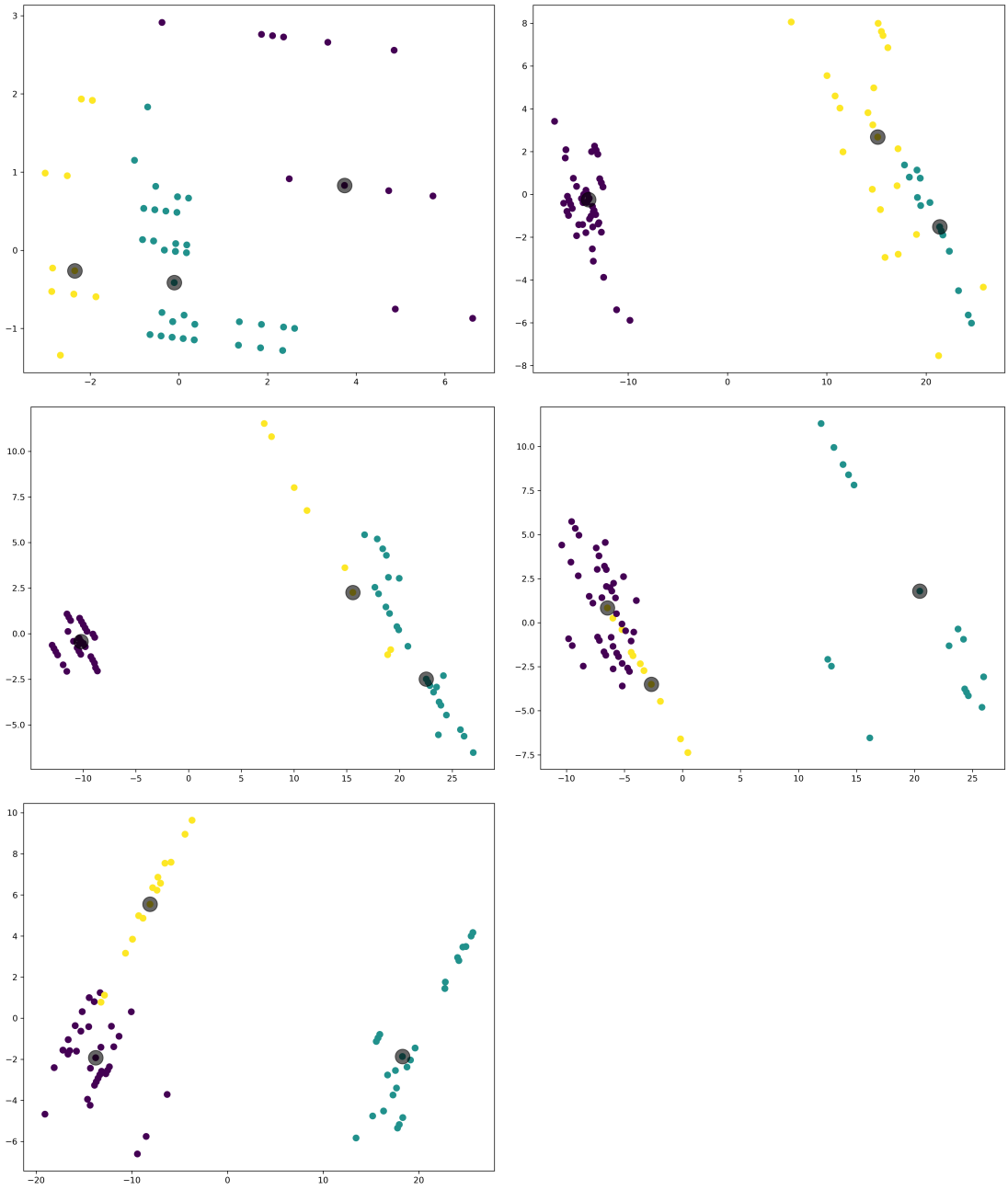
## 3.4 PCA:lla täydennetty EM-GMM -klusterointi



**Kuva 3.15:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa B. Pääkomponenttianalyysiä käytetty.

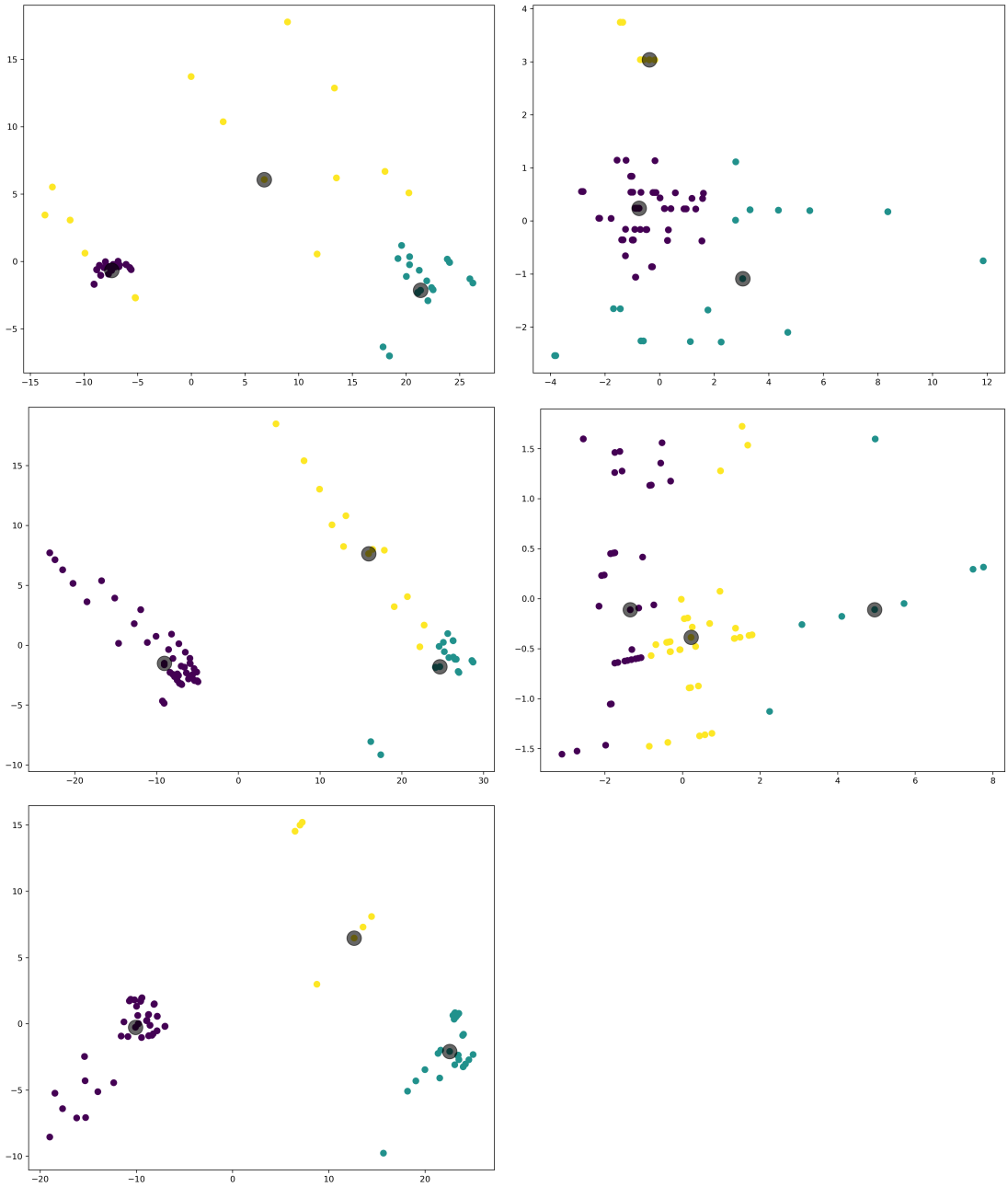
Kuvissa 3.15 ja 3.16 on kuvattu jokaisen aineiston osajoukon jakaminen kolmeen eri klusteriin sovittaen EM-algoritmia Gaussin sekoitemallille käyttäen pääkomponenttianalyysia. Kuvaajissa x-akseli kuvaa lämpötilaa A ja y-akseli kuvaa lämpötilaa B. Jokaisessa

kuvaajassa näkyy kolme klusterikeskusta, jotka on merkitty harmaalla ympyrällä.



**Kuva 3.16:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa B. Pääkomponenttianalyysiä käytetty.

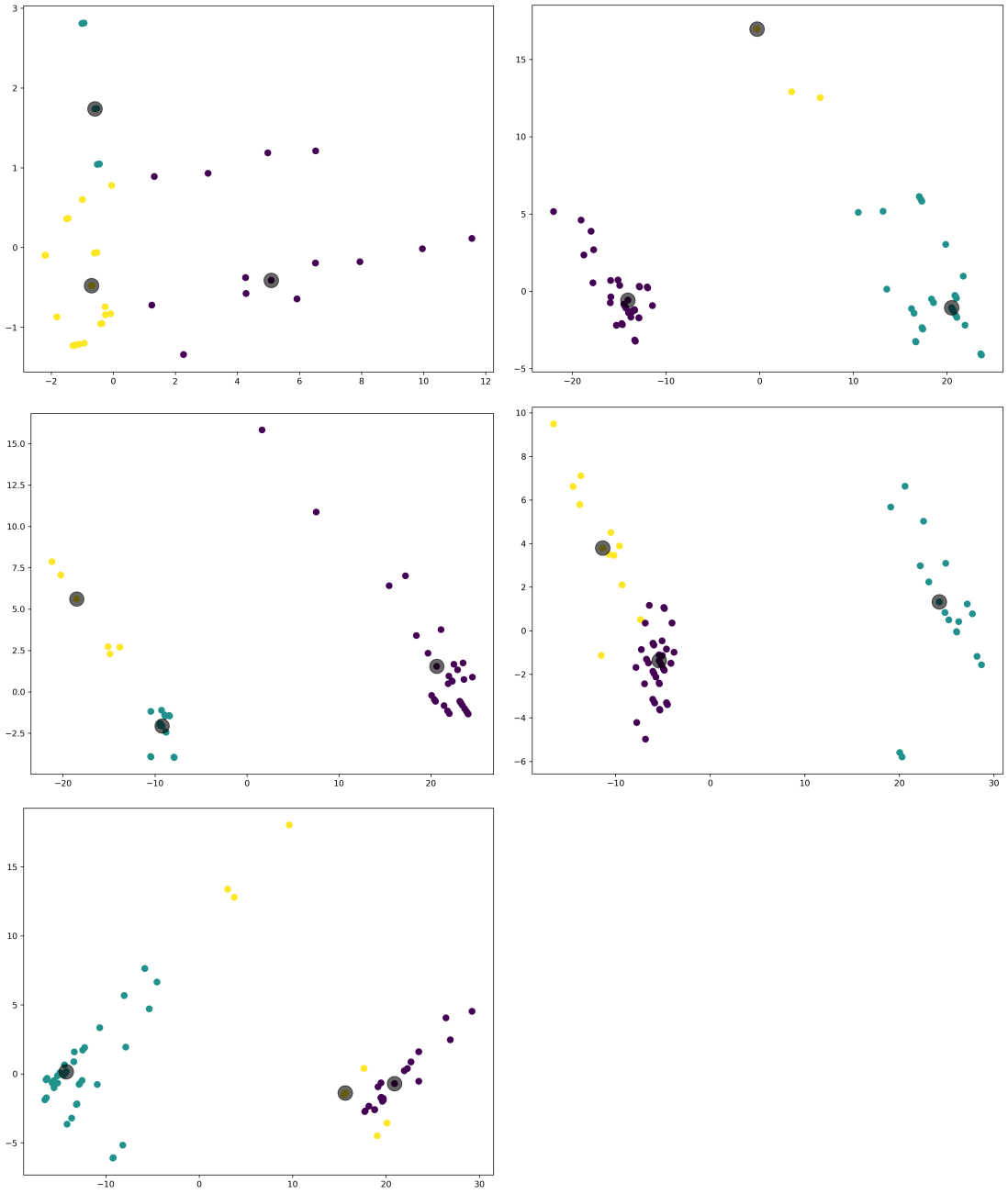
EM-GMM -klusterointi ilman PCA:ta ja käyttäen PCA:ta eroavat hieman toisistaan näissä tapauksissa. Kuvan 3.16 toisessa, kolmannessa ja neljännessä kuvaajassa EM-GMM klusteroinnilla ilman PCA:ta saadaan selkeämmät klusterit. Vastaavasti kuvan 3.15 kolmannessa kuvaajassa EM-GMM klusteroinnilla käyttäen PCA:ta saadaan selkeämmät klusterit. Muissa kuvaajissa molemmissa menetelmissä klusterit erottuvat yhtä selkeästi tai epäselvästi.



**Kuva 3.17:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa B ja y-akseli lämpötilaa C. Pääkomponenttianalyysiä käytetty.

Kuvissa 3.17 ja 3.18 on kuvattu jokaisen aineiston osajoukon jakaminen kolmeen eri klusteriin sovittaen EM-algoritmia Gaussin sekoitemallille käyttäen pääkomponenttianalyysia. Kuvaajissa x-akseli kuvaa lämpötilaa B ja y-akseli kuvaa lämpötilaa C. Jokaisessa kuvaajassa näkyy kolme klusterikeskusta, jotka on merkitty harmaalla ympyrällä.

EM-GMM -klusterointi ilman PCA:ta ja käyttäen PCA:ta eroavat hieman toisistaan näissä tapauksissa. Kuvan 3.17 toisessa ja kuvan 3.18 viidennessä kuvaajassa EM-GMM klusteroinnilla ilman PCA:ta saadaan selkeämmät klusterit. Muissa kuvaajissa molem-

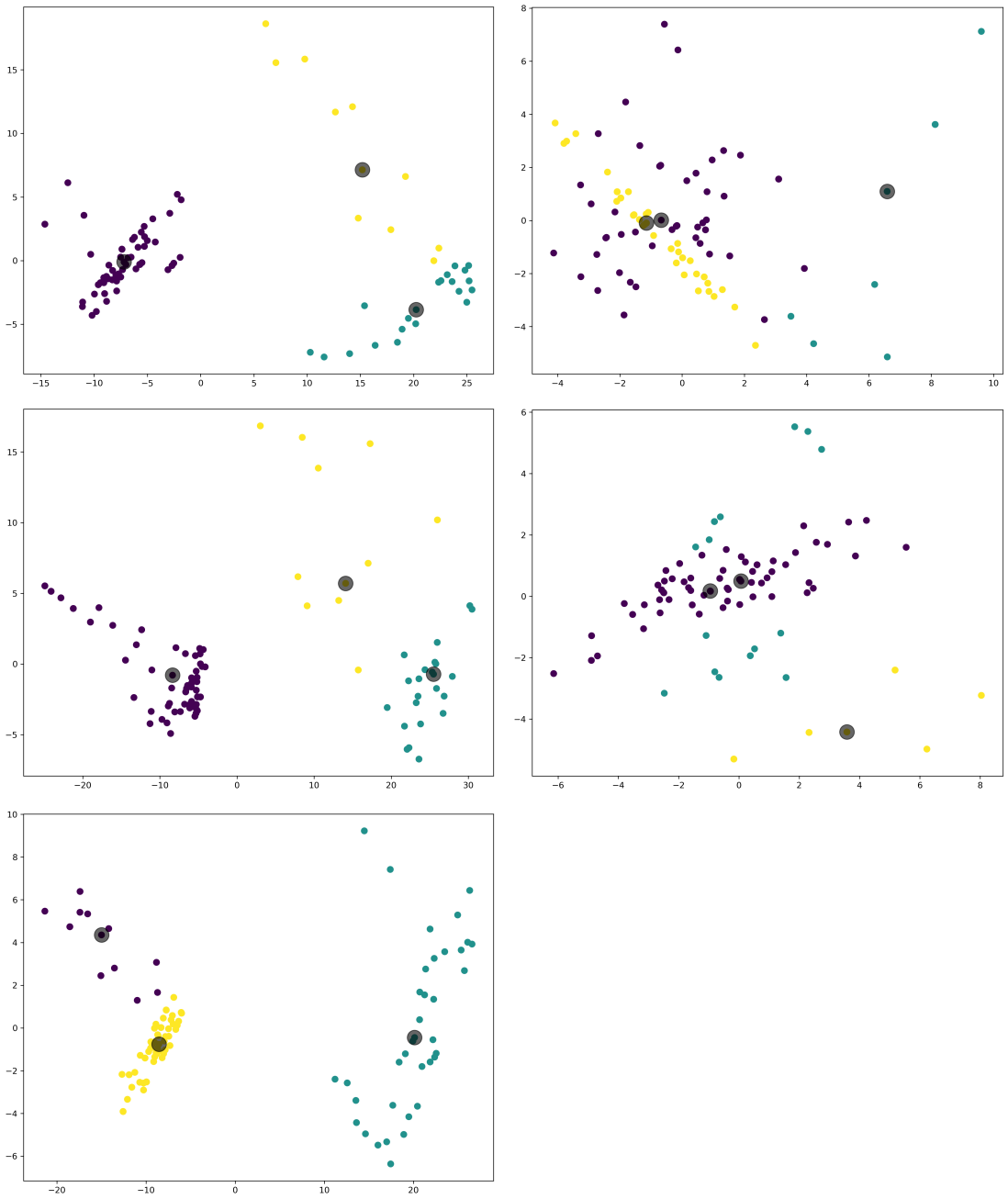


**Kuva 3.18:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa B ja y-akseli lämpötilaa C. Pääkomponenttianalyysiä käytetty.

missa menetelmissä klusterit erottuvat yhtä selkeästi tai epäselvästi.

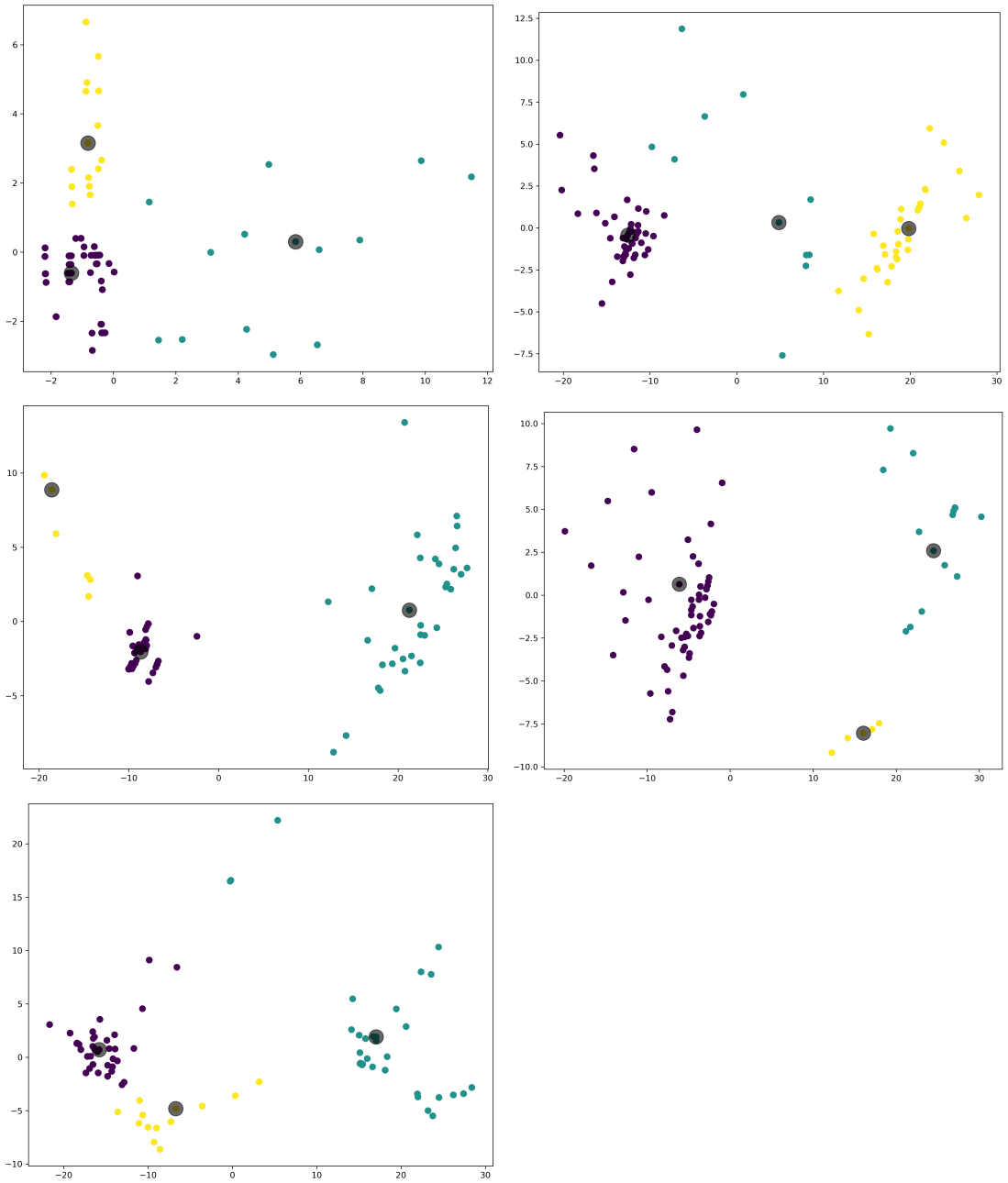
Kuvissa 3.19 ja 3.20 on kuvattu jokaisen aineiston osajoukon jakaminen kolmeen eri klusteriin sovittaen EM-algoritmia Gaussin sekoitemallille käyttäen pääkomponenttianalyysia. Kuvaajissa x-akseli kuvaa lämpötilaa A ja y-akseli kuvaa lämpötilaa C. Jokaisessa kuvaajassa näkyy kolme klusterikeskusta, jotka on merkitty harmaalla ympyrällä.

EM-GMM -klusterointi ilman PCA:ta ja käyttäen PCA:ta ei eroa juuri ollenkaan näissä tapauksissa. Molemmissa menetelmissä klusterit erottuvat yhtä selkeästi tai epä-



**Kuva 3.19:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa C. Pääkomponenttianalyysiä käytetty.

selvästi.



**Kuva 3.20:** EM-GMM -klusterointi, jossa x-akseli kuvaa lämpötilaa A ja y-akseli lämpötilaa C. Pääkomponenttianalyysiä käytetty.

## 3.5 Suhteellinen muutos

Lasketaan lämpötilan C ja lämpötilan B muutos sekä muutoksen suhde. Näin ollen pystymme suhteellisen muutoksen avulla laskemaan, pystyisikö lämpötilaa A muuttamaan.

Lasketaan kuinka paljon sellaisia mittauspisteitä löytyy, joissa uusi ehdotettu lämpötila A' on suurempi kuin oikea lämpötila A, jolloin olisi mahdollista nostaa kyseistä lämpötilaa. Näitä mittauspisteitä löytyi yhteensä 242. Seuraavaksi lasketaan kuinka paljon sellaisia mittauspisteitä löytyy, joissa uusi ehdotettu lämpötila A' on pienempi kuin oikea lämpötila A, jolloin kyseistä lämpötilaa ei ole mahdollista nostaa. Näitä mittauspisteitä löytyi 863. On kuitenkin otettava huomioon saman mittauskerran eri pudotukset ja tuotteet, sillä mikäli yhdellä mittauskerralla on esimerkiksi vain yksi tuote, jonka perusteella olisi järkevää nostaa lämpötilaa A, tätä ei kannata kuitenkaan tehdä, sillä lämpötilan nostaminen vaikuttaa myös muihin tuotteisiin ja tässä tilanteessa niiden perusteella ei lämpötilaa A kannata nostaa.

Valitaan datasta vain jokaisen reitin pienin ehdotus uudelle lämpötilalle A', näin pystymme tarkastamaan onko järkevää nostaa lämpötilaa A. Näistä vielä valitaan ainoastaan ne, joiden uusi ehdotettu lämpötila A' on suurempi kuin oikea lämpötila A. Tässä tilanteessa saadaan vain 74 sellaista mittauspistettä ja näistäkin osassa on mittausvirheitä. Tämä olisi siis $\frac{74}{1105} \times 100\% \approx 6,7\%$ aineistosta, joka on todella pieni osa alkuperäisestä aineistosta ja näin ollen siitä ei pystytä tekemään tilastollisesti merkitsevää ratkaisua.

# 4. Pohdinta

Tutkielmassa käytetyt mittauskerrat jäivät vain 467 mittaukseen, jonka vuoksi lopullinen aineisto jäi melko suppeaksi. Tämän lisäksi aineistosta jäi puuttumaan muutamia tietoja. Näiden vuoksi kaikkiin tutkimuskysymyksiin ei pystytty vastaamaan, mutta muutamaan tutkimuskysymykseen kuitenkin saatiin vastaukset.

Pääkomponenttianalyysin soveltaminen EM-GMM klusteroinnissa ei tuottanut toivottua tulosta klusterointiin. Myöskään ei löytynyt selvää eroa $k$-means klusteroinnin ja EM-GMM lähestymistavan välillä. Osassa mittauksia $k$-means algoritmi tuotti parempia tuloksia, mutta oli myös tapauksia, joissa EM-GMM lähestymistapa tuotti parempia tuloksia. Tämänkaltaisessa tutkimuksessa $k$-means -klusterointi toimi kuitenkin paremmin.

Tutkimusta pystyisi jalostamaan vielä eteenpäin ja puuttuviin tutkimuskysymyksiin saisi mahdollisesti vastauksen, jos suoritetaan lisää mittauksia ja liitetään ne tämän tutkielman aineistoon. Tällöin pystyttäisiin vastaamaan myös tarkemmin tutkimuskysymyksiin, joihin saatiin vastaus myös tämän tutkimuksen avulla.

# Lähteet


[1] WikiHow calculate percentage error. https://www.wikihow.com/Calculate-Percentage-Error. Updated: January 23, 2020.

[2] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[3] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[4] D. A. Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741, 2009.

[5] V. Ritvanen, A. Inkiläinen, A. von Bell, and J. Santala. *Logistiikan ja toimitusketjun hallinnan perusteet*. Suomen Osto- ja Logistiikkayhdistys LOGY ry, Ratamestarinkatu 7 A, 00520 Helsinki, 2011.

[6] L. Törnqvist, P. Vartia, and Y. O. Vartia. How should relative changes be measured? *The American Statistician*, 39(1):43–46, 1985.

[7] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584, 2001.

[8] M.-S. Yang, C.-Y. Lai, and C.-Y. Lin. A robust em clustering algorithm for gaussian mixture models. *Pattern Recognition*, 45(11):3950–3961, 2012.